AFRL-RI-RS-TR-2011-169

CROSS LAYERED MULTI-MESHED TREE SCHEME FOR COGNITIVE NETWORKS

SPECTRACOM CORPORATION

JUNE 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2011-169 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

MICHAEL J. MEDLEY
Work Unit Manager

/s/

PAUL ANTONIK, Technical Advisor
Advanced Computing Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUN 2011	Final Technical Report	FEB 2008 – NOV 2010

4. TITLE AND SUBTITLE

CROSS LAYERED MULTI-MESHED TREE SCHEME FOR COGNITIVE NETWORKS

5a. CONTRACT NUMBER
FA8750-08-C-0061

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Nirmala Shenoy

5d. PROJECT NUMBER
AN08

5e. TASK NUMBER
RI

5f. WORK UNIT NUMBER
TS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Spectracom Corporation
c/o John Fischer
95 Methodist Hill Drive
Rochester, NY 14623

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGF
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2011-169

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. PA# 88ABW-2011-2961
Date Cleared: 25 May 2011

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This work was conducted to demonstrate the strengths of integrated protocol stacks towards addressing the heterogeneous airborne network scenarios. The proposed solution was implemented using Opnet simulation tool and evaluated for various ground and airborne tactical networks. Results indicate the superior performance of the proposed solution.

15. SUBJECT TERMS
Cross-layer optimization, intra-cluster routing, packet forwarding, inter-cluster routing, mesh network communications, route spoofing, dynamic spectrum sensing, adaptive network communications

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			MICHAEL J. MEDLEY
U	U	U	UU	54	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

Summary 1

1 Introduction - Cognitive Airborne Networks 2

2 Methods Assumptions and Procedures 3

2.1 The Multi-Meshed Tree (MMT) Algorithm 3

2.2 Justification 7

2.2.1 Related Work 7

2.2.2 MMT Algorithm Capabilities and Operational Improvements 8

3 Progress against Planned Objectives 10

4 Results and Discussions 10

4.1 Surveillance Networks With Omni-Directional Antennas 10

4.2 Peer-to-Peer Communications Networks For Ground Troops 19

4.3 Peer-to-Peer Airborne Backbone Networks With Omni-Directional Antennas 20

4.4 Surveillance Networks with Directional Antennas 21

4.4.1 Comparison of Two Link Optimization Strategies 21

4.4.2 Optimizing Link Assignment to MMT Cluster Parameters 30

4.5 Peer-to-Peer Airborne Backbone Networks with Directional Antennas 35

4.5.1 Hybrid Scheduler and Link Assignment Based on MMT 35

4.5.2 Distributed Scheduler and Inter-Cluster Communications 38

5 Conclusions 43

6 Bibliography 44

List of Acronyms 47

List of Figures

Figure 1 Notional Airborne Network Topology 2
Figure 2 Meshed Trees 4
Figure 3 Cluster Formation using VIDs 4
Figure 4 Proactive Routing Based on VIDs 5
Figure 5 Overlapped Clusters 5
Figure 6 Meshed Tree-Based Reactive Routing 6
Figure 7 Frame Forwarding in Meshed Tree Scheme 6
Figure 8 Integrated MMT-Based Solution 9
Figure 9 Cluster Formation Based on Meshed Trees 11
Figure 10 Overlapped Clusters Based on MMT 12
Figure 11 Traffic Forwarding Based on VIDs 13
Figure 12 Setting a Data Session and Implicit Acks 14
Figure 13 Use of NCTS Mode for Collision Avoidance 14
Figure 14 Session on Wait Time Calculations 14
Figure 15 Data Retry Model 14
Figure 16 Performance Graphs – 20-Node Scenario (surveillance networks/omni-directional) 16
Figure 17 Performance Graphs – 50-Node Scenario (surveillance networks/omni-directional) 17
Figure 18 Performance Graphs – 100-Node Scenario (surveillance networks/omni-directional) 18
Figure 19 Performance Graphs – 20-Node Scenario (peer-to-peer/ground troops) 19
Figure 20 Performance Graphs – 20-Node Scenario (airborne backbone/omni-directional) 20
Figure 21 Performance Graphs – 50-Node Scenario (airborne backbone/omni-directional) 21
Figure 22 Scheduler Operation with Other Modules 24
Figure 23 Operational Sequence in Scheduling 25
Figure 24 Sample schedule for Cluster in Figure 9 27

Figure 25 Performance Graphs – 20-Node Scenario (surveillance/link assignment strategies) 28
Figure 26 Performance Graphs – 50-Node Scenario (surveillance/link assignment strategies) 29
Figure 27 Performance Graphs – 100-Node Scenario (surveillance/link assignment strategies) 30
Figure 28 Performance with Varying Cluster Sizes (surveillance/directional)..... 31
Figure 29 Performance to Varying Number of Time Slots (surveillance/directional)..... 32
Figure 30 Performance to Varying Number of Time Slots (fixed frame size) 32
Figure 31 Performance Graphs – 20-Node Scenario (surveillance/directional) 33
Figure 32 Performance Graphs – 50-Node Scenario (surveillance/directional) 34
Figure 33 Performance Graphs – 100-Node Scenario (surveillance/directional) 34
Figure 34 Sample Schedule in Meshed Tree Clusters 36
Figure 35 Performance Graphs – 20-Node Scenario (airborne backbone/directional)..... 37
Figure 36 Performance Graphs – 50-Node Scenario (airborne backbone/directional)..... 37
Figure 37 Performance Graphs – 100-Node Scenario (airborne backbone/directional)..... 38
Figure 38 Reactive Routing Based on Meshed Trees Clusters..... 39
Figure 39 Performance Graphs – 20-Node Scenario (airborne backbone/distributed scheduler)..... 41
Figure 40 Performance Graphs – 50-Node Scenario (airborne backbone/distributed scheduler)..... 42
Figure 41 Performance Graphs – 100-Node Scenario (airborne backbone/distributed scheduler)..... 42

List of Tables

Table 1 VID table at the CH 22
Table 2 Opnet Simulation Parameters 27
Table 3 Sample Schedule Generated by the Distributed Scheduler 40

Summary

The *Department of Defense* is engaged in efforts to develop an IP-based *Airborne Network* (AN) to interconnect several mobile airborne platforms. The *Airborne Network* will consist of cognitive AN nodes that advertise their identity and location for discovery by other AN nodes. This will help in establishing connections among nodes that are airborne, in space, or on the surface.

From a network-centric point of view two of the protocols that are significantly impacted by the dynamic topology and link connectivity challenges posed by MANETs such as Airborne Networks are the '*routing'* and '*medium access control'* (MAC) protocols. Such protocols exchange 'control' messages to maintain network connections and update routes. As the dynamics and size of the network increase, the number of control messages exchanged increases. Scalability in MANETs is addressed through two techniques, one is 'clustering,' which introduces hierarchy in the MANET and restricts most communications and data dissemination to be within a cluster (intra-cluster) and uses inter-cluster communications only when required. The second is hybrid routing, which restricts proactive route maintenance within a given zone and adopts reactive routing when communicating with nodes outside the zone. Algorithms to achieve functions like clustering, or zone-defining, proactive and reactive routing can be complex and very often a different algorithm and/or protocol is used for each scheme. This requires interworking among these algorithms and protocols, which adds to overhead and results in complex solutions. Use of IP layer to support routing functions makes algorithms and protocols layer 3 dependent (for example IPv4, IPv6 or others). Layer 3 protocols and IP addresses are required only when communicating across subnets. Continuing use of two addresses such as the MAC and IP addresses for communications within a subnet can introduce a higher overhead in computing and bits transmitted, while increasing the system size and reducing battery life.

In this project a compact protocol stack that combined clustering, routing, and MAC functions operate at layer 2 using a single algorithm called the Multi-Meshed Tree (MMT) Algorithm. MMT algorithm provides the robustness and redundancy inherent in mesh topologies and uses the tree branches to forward packets. MMT is a single algorithm that is used to form *multiple multi-hop* clusters while providing *multiple* proactive routes to cluster clients within the cluster as well as an extension that supports reactive routing. All functions are achieved without flooding or requiring topology information and the operation of routing and MAC were achieved using a single address. The MMT-based compact protocol stack has been evaluated for different Airborne network scenarios.

1 Introduction - Cognitive Airborne Networks

The *Department of Defense* is engaged in efforts to develop an IP-based *Airborne Network* (AN) to interconnect several mobile airborne platforms. The *Airborne Network* will consist of cognitive AN nodes that advertise their identity and location for discovery by other AN nodes. This will help in establishing connections among nodes that are airborne, in space, or on the surface. The AN nodes, however, vary in their communication capability needs, flight patterns, and size, weight and power constraints. AN nodes are expected to perform different functions that can be broadly categorized into relaying (receive and transmit with the same data formats and on the same media/frequency), translating (receive and transmit with the same data formats but on different media or frequencies), or gateway (receive and transmit with different data formats and on different media/frequencies) nodes.

Based on their functional capabilities, AN nodes will be used to

- establish connection and inter-network with prearranged static and ad hoc subnets (with a capability to leave or join a subnet any time) and with specific sets of nodes;
- establish connections to legacy or IP network; and
- perform routing or switching of IP packets to/from an IP-based space or terrestrial subnet or backbone.

Figure 1 shows a notional *Airborne Network* topology using AN nodes with typical functions expected from these nodes described beside them.

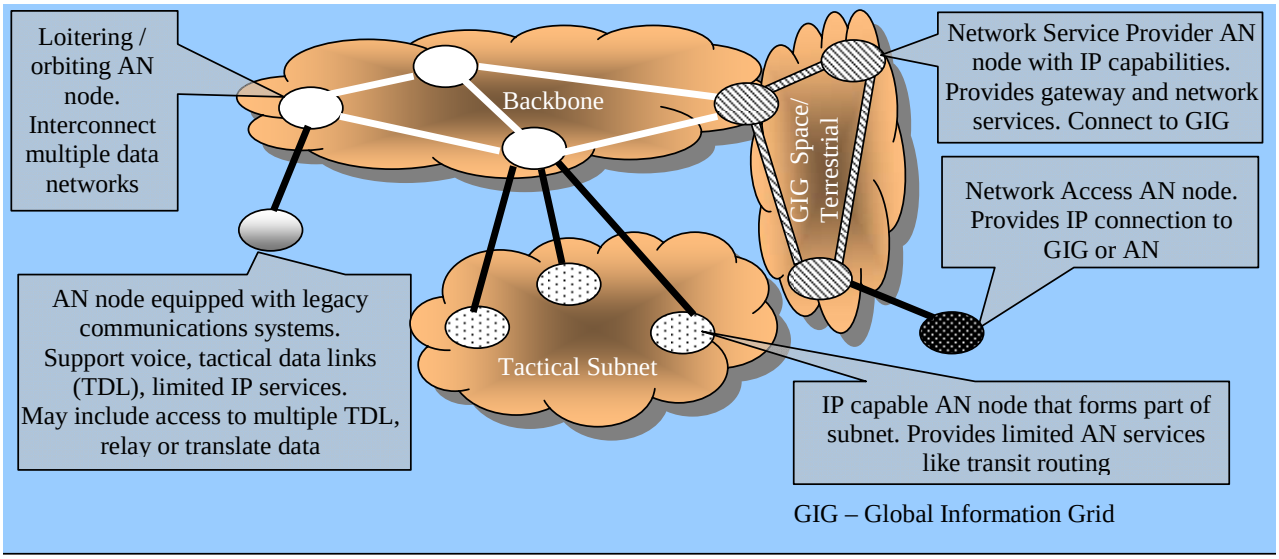

Figure 1 Notional Airborne Network Topology

In Figure 1, the core set of loitering and/or orbiting AN nodes that form the network's backbone are capable of hosting a set of heterogeneous high capacity "quasi-persistent" wireless links. These links can be used to provide redundant high bandwidth connections to interconnect other networks and AN subnets. The AN nodes are also expected to provide AN and/or Global Information Grid (GIG) services such as directory and gateway services.

It is desired to have a solution for the AN that

- connects heterogeneous subnetworks;
- provides robust connectivity;
  - under highly dynamic conditions;
  - with rapidly changing network topologies;
- alleviates the impact of changing connectivity as links enter and leave the networks;
- enables Quality of Service (QoS) via traffic prioritization;

- makes optimal use of the resources like bandwidth and available links;
- enhances security;
- coexists with IPv4 and IPv6; and
- has cognitive capabilities to intelligently accommodate application demands while taking into consideration the physical layer constraints.

2 Methods Assumptions and Procedures

From a network-centric point of view two of the protocols that are significantly impacted by the dynamic topology and link connectivity challenges posed by MANETs such as Airborne Networks are the '*routing'* and '*medium access control'* (MAC) protocols. Such protocols exchange 'control' messages to maintain the network connections and update routes. As the dynamics and size of the network increase, the number of control messages exchanged increases. However it is very essential to keep them low so that more of the bandwidth is available for transporting payload. Below we state some accepted approaches for scalability, followed by some desirable features to simplify system complexity and configurability.

Scalability: To address scalability in MANETs, two popular techniques are adopted. One is 'clustering' which introduces hierarchy in the MANET and restricts most communications and data dissemination to be within a cluster (intra-cluster) and uses inter-cluster communications only when required. The second is hybrid routing, which restricts proactive route maintenance within a given zone and adopts reactive routing when communicating with nodes outside the zone. This approach reduces route discovery overhead considerably as routes to distant nodes are discovered on a 'need to communicate' basis. The drawback of 'lead' latencies and rediscovery of routes in the event of route breaks, however, is a continuing issue.

Network algorithm and protocol complexity: Algorithms to achieve functions like clustering, or zone-defining, proactive and reactive routing can be complex and very often a different algorithm and/or protocol is used for each scheme. This requires interworking among these algorithms and protocols, which adds to overhead and results in complex solutions. Use of IP layer to support routing functions makes algorithms and protocols layer 3 dependent (for example IPv4, IPv6 or others). Layer 3 protocols and IP addresses are required only when communicating across subnets. Continuing use of two addresses such as the MAC and IP addresses for communications within a subnet can introduce a higher overhead in computing and bits transmitted, while increasing the system size and reducing battery life.

Configurability: Most MANET protocol functions are governed by the applications and services supported. Application-specific configurability in the protocols would be very desirable.

2.1 The Multi-Meshed Tree (MMT) Algorithm

The solution developed at RIT, uses a ***single novel algorithm*** that is able to

- Form multiple multi-hop clusters of configurable sizes;
- Set up proactive routes without flooding as the clusters are formed; and
- Discover reactive routes between distant nodes without flooding discovery messages. The route dependency between the distant nodes is reduced to the number of clusters between them and not the actual forwarding nodes. The reactive routes are concatenations of proactive routes and are rarely stale.

The Concept: Consider Figure 2 shown below. Two tree branches are shown originating at node 'A', which is the root node. We call such a tree a meshed tree as the two branches that originate at 'A' mesh as shown. However each branch is maintained without loops even though it may seem that the tree branches are meeting at some particular node, e.g., nodes J and K. Packet flows to J and K will not lead to loops as they follow paths as per the 'IDs' noted in the branches. The meshed tree creation is thus possible because of the novel numbering scheme. In the sections that follow, we describe in detail the different components of the MMT algorithm and their operational aspects.

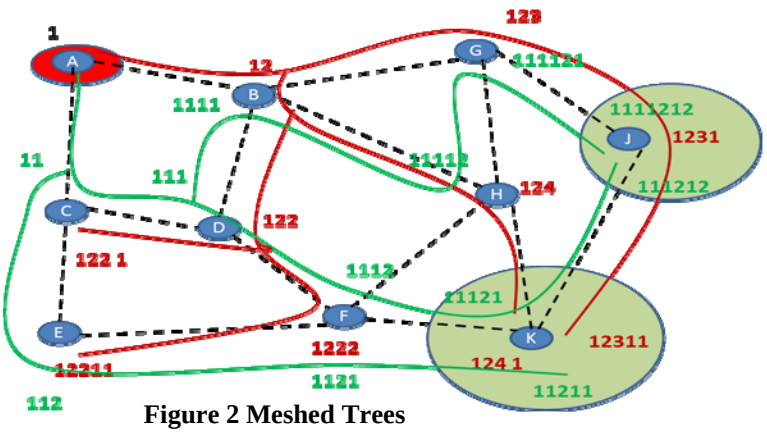

Figure 2 Meshed Trees

Multiple multi-hop overlapped cluster formation within the MMT model requires cluster head election and, subsequently, cluster formation.

Cluster head election involves determining a suitable cluster head in a locality using criteria like node IDs, credentials, power level, MAC address and the number of neighbors [45] or combinations thereof [40-43]. *We adopt one such algorithm* that is based on the number of neighbors and use IDs to resolve a tie. This election process is required only when all nodes are deployed at the same time or when there is need for resolution. Once elected, a cluster head continues for a predefined period or till it is disabled or dies [39].

Cluster formation is the process by which nodes decide to join a cluster head for reasons like better signal and so on. Most cluster formation involves nodes at one-hop joining a suitable cluster head. In the K-hop clusters [46], nodes at 'K' hops from the cluster head connect to the desired cluster head; their connectivity is maintained using distance vector routing, spanning tree or variations. The MMT cluster formation algorithm is *different* as it allows the cluster head to decide which of the requesting cluster clients will be in its cluster based on a defined cluster size and hops from the cluster head.

MMT Cluster Formation: In Figure 3, the first picture assumes that node A with a unique ID (UID) = 100 is elected cluster head. 'A' advertises its UID as a cluster head virtual ID (VID). Node B hears the advertisement and sends a join request to 'A'. 'A' will allocate a VID = 1001 to B. A parent node is allowed a maximum of 9 one-hop children to reduce traffic bottleneck at the parent node. When node A accepts a child, it will allocate the child a VID that is its own VID appended with a single digit integer. If another nodes wants to join as a first hop child of cluster head A it will be given a VID 1002, the following one will be given a VID = 1003 and so on.

In the second picture in Figure 3, B has acquired a VID and now advertises this VID. Node C hears B and sends a join request to B. B follows a similar procedure in allocating a VID to C and C gets VID = 10011. C then registers with the cluster head 'A'. In the registration request C provides its UID and its newly acquired VID. The path taken by the registration request will be C->B->A, so that the parent is aware of the registration. Acceptance into cluster is completed by cluster head A sending 'registration accept'. The cluster head can advertise the cluster size in the advertisement messages, so that clients will not accept any more new children into the cluster.

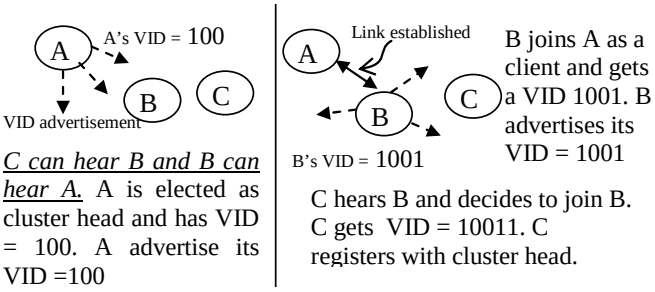

Figure 3 Cluster Formation using VIDs

The VIDs directly provide the number of hops from the cluster head. The cluster head ID is inherent in all VIDs used by the cluster clients. The VIDs simplify the process of multi-hop cluster creation with very low overhead both for cluster formation and maintenance. If a cluster head dies, its one-hop children (which can be identified from the VIDs) that are in hearing range of one another resolve to be cluster head and through bulk registration and deregistraion form new clusters. The concept of bulk registration is new to research in clustering and made possible because of the inherent information in the VIDs.

Multiple Proactive Routes in MMT Clusters: Using Figure 4, we explain multiple proactive route establishment *as the clusters are formed.* As shown within the shaded circle, the one-hop children of node 'A' are assigned VIDs whose prefixes indicate their origin at A, e.g., B is assigned VID = 1003, C is assigned VID = 1001, and D is assigned VID = 1002. The two-hop children are K, G, H and J, which have respectively VIDs 10012, 10031, 10043

and 10053, which have been derived from their parents namely C, B, F and E respectively. Note that the VIDs carry the route information from the cluster clients to the cluster heads. The shaded nodes in Figure 2 have multiple VIDs. The secondary VIDs were acquired by overhearing the advertisements from neighbors and joining as their children. Multiple VIDs thus result in multiple routes.

Due to mobility, if a node loses one VID, it can fallback on the other backup routes. For example in Figure 4, assume J moves in the direction indicated by the dashed arrow. It may lose connectivity with E, but it still has connectivity to the cluster via H. Nodes continually overhear activity by their neighbors and acquire new and better VIDs (fewest hops for example, with fewest digits in the VID), thus eliminating the possibility for stale routes. The underlying principle is based on 'source routing' which has been used in reactive routing schemes; *we use the same principle to set up multiple proactive routes to the cluster head from the cluster client.* The process has been very much *simplified and is an integral part of the cluster formation. No routing tables or states are required at cluster clients as route information is carried in the VIDs.* The dynamic multiple proactive routes establishment provides robust connectivity with low overhead and is the first of its kind to the best of our knowledge. The VIDs indicate a branch from the cluster head in the cluster. Multiple VIDs help in meshing the tree branches. *No complex computations are required to avoid loops in the mesh*; a node simply checks its VIDs and compares the integers after the cluster head VID to determine if a loop will be formed if it were to request a particular VID.

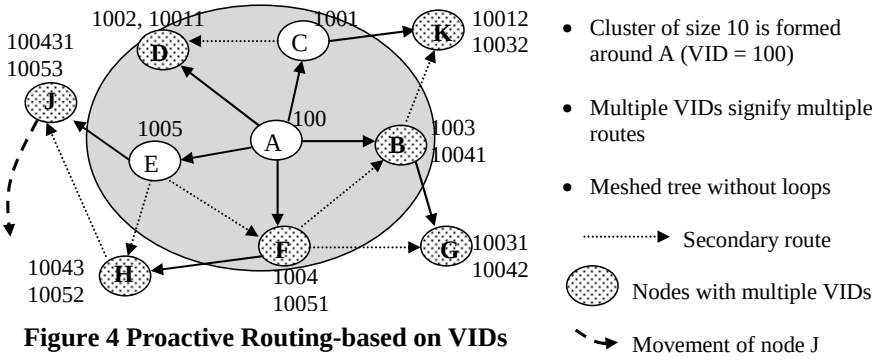

Figure 4 Proactive Routing-based on VIDs

Overlapping Clusters in MMT: To improve route robustness in the scheme, the clusters are allowed to overlap. In Figure 5, there are two clusters, one with A as the cluster head and the other with L as the cluster head. The client VIDs under A start with '100', whereas the client's VIDs under L start with '109', (the cluster head VID is underlined). In the figure, nodes B, K, P and G with double circles are members of both clusters. If the mobility pattern of these nodes leads them to move towards cluster head L, they may lose their VIDs with 'A' but will be connected to cluster 'L' and vice versa. The overall connectivity in the network is thus enhanced, through the ***'multi-meshed tree'*** concept.

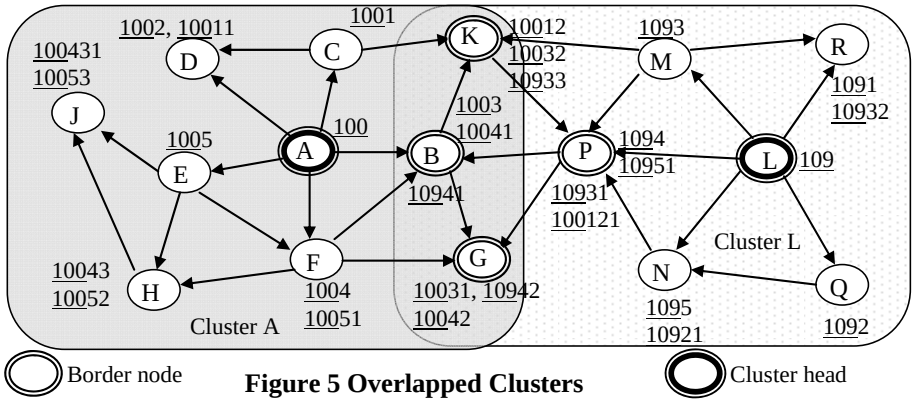

Figure 5 Overlapped Clusters

Overlapping clusters are easily created with the MMT algorithm. Nodes that belong to more than one cluster register their multiple VIDs and UID with the cluster heads. This leads to some very useful knowledge; 1) the cluster heads are aware of the neighboring cluster heads and their VIDs; 2) the border node knowledge can be used for inter-cluster communications; 3) reduction in route discovery flooding for inter-cluster communications.

MMT Reactive Routing and Route Discovery: Figure 6 shows four overlapping clusters. We explain reactive routing when node J (in cluster A) wants to communicate with node Q1 (in cluster L1). Node J sends route discovery message (with Q1/1204) to its cluster head A. Cluster head A notes that the destination is not in its cluster, it will record its VID in the "route record" field of discovery message and forward to neighboring clusters via B (border node between cluster A/100 and L/109) and H (border node between cluster A/100 and A1/105). Border nodes B and H will forward to the cluster heads L and A1, L respectively. As the Q1 destination does not lie in their respective cluster, A1 and L will forward the discovery message to their neighboring clusters after recording the VIDs in the 'route record' field. Two route discovery messages may finally reach cluster head L1/120. L1 forwards the route discovery message to Q1/1204. Recorded route will be 'A, A1, L1' and 'A, L, L1' – i.e. the cluster head VIDs. Route reply from Q1 to J follows the recorded path in reverse as *identified by the clusters.* The length of the 'recorded route' is that of the clusters along the way and not all forwarding nodes. So, the route between J and Q1 is dependent on only the change in the 3 cluster heads. The probability of route failure hence is reduced considerably.

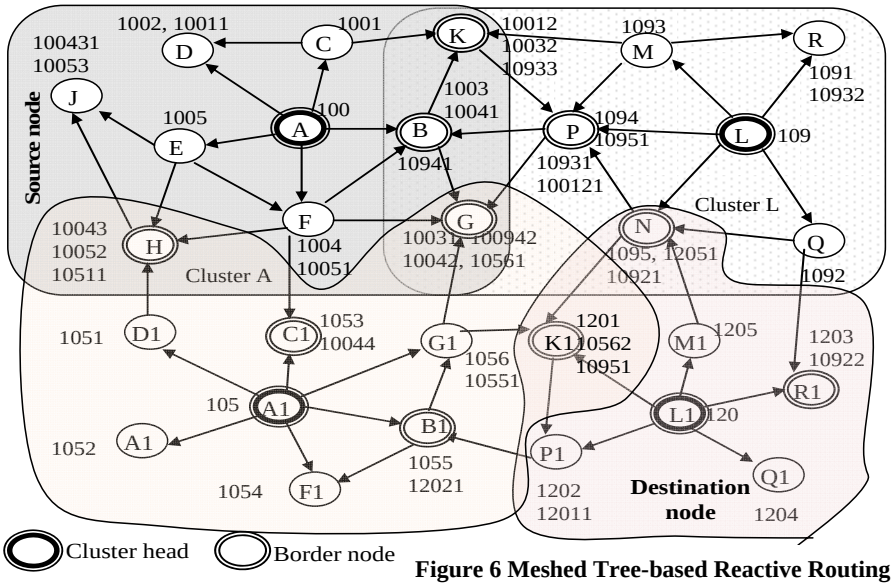

Figure 6 Meshed Tree-based Reactive Routing

Redirection Capability of MMT VIDs: In Figure 6 after receiving the route discovery message from 'A', let node P move away and lose its VID 1094. However it is aware that the route discovery packet is to be delivered to cluster head L. It will use its VID 10931 to deliver the packet to the cluster head. This redirection capability can be used while forwarding data packets too and thus the scheme is highly resilient to node movement and varying link conditions. This may seem similar to 'cached routes' but the routes in this case have a high probability of not being stale as they are updated locally based on neighbor activity.

MMT at Layer 2*:* We now provide the rationale for operating the MMT algorithm at layer 2. The VIDs used by MMT can be used for frame forwarding, and avoid the necessity for MAC addresses (Figure 7). Assume that cluster head '100' receives a packet destined to Node 'X'. The cluster head will locate the VID of node X, which in this case happens to be 100234. It then encapsulates the payload as shown by setting the source VID to its own VID and the destination VID to '100234,' and transmits the packet. The packet will be picked up by node with VID 1002, as it recognizes from the destination VID that the destination node is a child node along its branch and resends. This is repeated until the frame eventually is received at node 'X'.

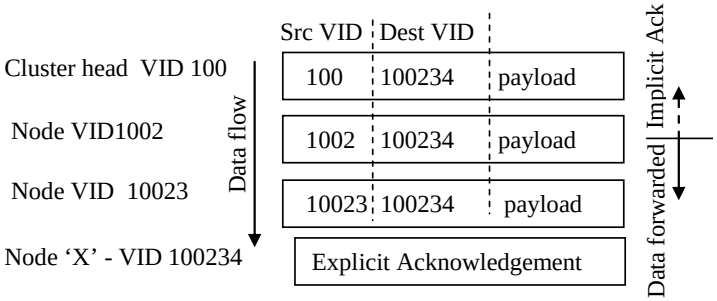

Figure 7 Frame Forwarding in Meshed Tree Scheme

Whether upstream or downstream, all nodes know when they are to forward a packet as the knowledge is implicit in the destination VID. The VID serves two purposes: identifying the route as well as identifying the next node to which the frame must be forwarded. As each forwarding node replaces its own VID in the source field, an *implicit advertisement* is issued. Lastly, as each node is aware of the next node that has to forward, the subsequent forwarding can be used as an *implicit acknowledgement* (except at the destination). Operating at layer 2 reduces the

routing function dependency on layer 3, which leads to fewer overhead bits, and hence faster processing as packets can be handled at layer 2. The scheme does not, however, preclude use of IP at layer 3.

What is Novel about MMT algorithm?

- Unification of tree and mesh topologies is novel. The algorithm provides the robustness and redundancy inherent in mesh topologies and uses the tree branches to forward packets.
- MMT is the first of its kind where a single algorithm is used to form *multiple multi-hop* clusters while providing *multiple* proactive routes to cluster clients within the cluster as well as an extension supporting reactive routing.
- The reactive routes are a concatenation of the proactive routes and, hence, rarely stale.
- Reactive routes use a 'loose source routing' concept carrying the cluster head VIDs and are not impacted by the dynamics of all the intermediate nodes that are forwarding.
- All the above functions are achieved without flooding or requiring topology information.
- For its operation, MMT uses a simple numbering scheme (VIDs) and local computations.
- No complex algorithms are involved; the VID scheme facilitates all the above.
- The VIDs carry the proactive route information, hence cluster clients do not need to maintain routing tables and states, except when communicating with nodes outside their cluster.
- Inherent in the VIDs is information to avoid loop formation.
- Inherent in the VIDs is the 'hop count' from the cluster head and its VID.
- A single address is used for both routing and forwarding.

2.2 Justification

Given the extent of research in the topic area of MANETs, it is important that we justify the contributions of the MMT algorithm. This section is devoted to work conducted in the areas of routing and clustering in MANETs, the two major approaches in MANETs that can address scalability and connectivity. Several survey articles [11-14] published on MANET routing and clustering are indicative of the continuing challenge that this topic poses. In the following paragraphs, we briefly describe some salient features of the two main categories of routing protocols, namely the proactive and reactive routing protocols followed by a discussion of schemes that have stemmed from these two basic concepts to address scalability, dynamic adaptation, robust connectivity, and configurability.

2.2.1 Related Work

Proactive Routing Protocols disseminate route discovery messages in a network regardless of demand and hence routes are available when client nodes require them [37]. Mobility of nodes in a MANET using proactive routing can result in frequent flooding of update information as the topology changes. In large networks, such transmissions consume most of the bandwidth. *Fisheye State Routing* (FSR) introduces multilevel fisheye scopes to reduce routing update overhead by reducing routing packet size and update frequency [25, 26] to remote nodes. *Fuzzy Sighted Link State* uses the optimal routing algorithm, *Hazy Sighted Link State* [29], and further reduces link state message dissemination. These multi-scope approaches work well when the network grows in terms of number of hops end-to-end but may not be effective as the density of the network increases. *Optimized Link State Routing* (OLSR) [18 - 20, 28] reduces control overhead of topology discovery messages by using selected one-hop nodes as 'multi-point relays'. *Topology Broadcast Reverse Path forwarding* (TBRPF) [27] propagates link-state updates in the reverse direction on a spanning tree formed from all nodes to the source node.

Reactive Routing Protocols discover routes to destination nodes only when needed hence there is route discovery latency. Source nodes cache several such discovered routes. Routing overheads in reactive routing protocols could be considerably low as they are primarily due to route discovery and maintenance of the routes in use. A flooding control scheme during route discovery could enhance efficiency of reactive routing. At heavy traffic with large number of destinations, more sources will search for destinations using up the limited bandwidth. As mobility increases, route caching also becomes ineffective as pre-discovered routes break down, requiring repeated route discoveries [30]. *Dynamic Source Routing* (DSR) [16] a popular reactive routing protocol for MANETs, requires each packet to carry the full address of every hop in the route, from source to the destination, and hence faces scalability problems. *Ad Hoc On-demand Distance Vector* (AODV) [15] routing uses periodic beacon messages and sequence numbering procedure of *Destination Sequence Distance Vector* (DSDV) [17] as well as a route discovery scheme similar to DSR, with intermediate nodes maintaining the forwarding information. *Temporally Ordered Routing Algorithm* (TORA) [23] protocol uses link reversal, route repair and creation of *Directed Acyclic Graphs*

(DAGs), similar to *Light-Weight Mobile Routing* (LMR) [14] inheriting its benefits but reducing far-reaching control messages.

Hybrid Routing Protocols: Scalability in MANETs has been addressed by combining proactive and reactive routing in a hybrid approach, where the use of proactive routing is restricted to a limited area and reactive routing is used elsewhere[25]. Partitioning the MANET and introducing hierarchy to control disseminated messages also address scalability. *Sharp Hybrid Adaptive Routing Protocol* (SHARP) [37] is an application adaptive hybrid routing protocol that automatically finds the balance point between proactive and reactive routing. In SHARP, zones are automatically created around hot destinations. For reactive routing AODV or DSR can be used. SHARP *proactive routing protocol* combines DSDV and TORA, and uses an *update protocol.* Multi-path routing and local link repair enable robustness, low loss rate and predictable overhead. *Hybrid Routing for Path Optimality* [38] combines proactive route optimization to a reactive 'source' routing protocol to reduce average end-to-end delay in packet transmissions. Frequent route maintenance operations may generate higher routing overhead than a pure on-demand protocol. The *Zone Routing Protocol* (ZRP) [33] is a hybrid routing protocol, where each node has a pre-defined zone centered at itself. Any proactive routing within the zone and any on-demand routing for inter-zone communications could be used. To facilitate route discovery outside the zone a *Bordercast Resolution Protocol* that requires maintenance of the *Bordercast* tree which incurs high overheads is used. When network size increases, ZRP's behavior becomes similar to on-demand routing. ZRP maintains separate tables for proactive and reactive routes. *LANMAR* ad hoc routing protocol [34, 35] uses a local scope routing scheme based on FSR and elects landmark nodes to keep track of logical groups. To forward outside the scope, packets are routed towards the landmark in the destination's logical group. LANMAR uses truncated local routing table and "summarized" routing information to reduce overhead.

Hierarchical Routing: Routing table size and control overhead can be reduced considerably through hierarchical routing [21, 24]. Nodes geographically close to each other form clusters with a cluster head communicating to other nodes on behalf of the cluster. Different routing strategies can be used inside and outside the cluster. *Cluster head Gateway Switch Routing* [32] is a cluster-based hierarchical routing scheme that uses *distance vector* routing and maintains a cluster member table and a routing table at each node; such maintenance and complexity in clustering in a mobile environment introduces significant overhead. A mobile gateway node connects two or more clusters. *Hierarchical State Routing (HSR)* [32] is a multi-level, clustering-based link state routing protocol that uses the clustering scheme recursively. In HSR, *Hierarchical ID* (HID) a sequence of MAC addresses of nodes on the path from the top hierarchy to the node itself is used. Nodes dynamically and locally update their HIDs upon receiving updates. Continually changing HID make tracking of nodes difficult. HID registrations and translations require complex management. *Mobile Backbone Networks* (MBNs) [36] use hierarchy concept to form a higher level backbone network by utilizing special *backbone nodes* (BNs) with low mobility to have an additional powerful radio to establish wireless link among themselves. Multi-level MBNs can be formed recursively but introduce complexity. LANMAR [34, 35] was extended to route in the MBN. Redundant BNs are to be deployed to counter BN failures.

The MAC protocols for MANETs can be random access protocols or scheduled protocols, which use time division multiplexing or some other form of multiplexing techniques. The choice of MAC protocols also depends on the use of directional or omni-directional antennas in the AN nodes.

2.2.2 MMT Algorithm Capabilities and Operational Improvements

MMT Dynamic Adaptation and Robust Connectivity: The routes in MMT algorithm adapt and heal to changing topology locally. The multiple proactive route construction in MMT clusters does not depend on topology information, hence they are able to adapt quickly to topology changes. On the failure of a link, only the nodes downstream in a tree branch are affected. As there are multiple proactive routes, in the event of failure of one route, in MMT a node can immediately fallback on secondary routes, resulting in a convergence time of almost '0'. While a message is being forwarded to a cluster head, failure in routes do not affect data forwarding as a node can use one of its other VIDs to forward the message to the cluster head. We call this the redirection capability of MMT. MMT reactive routes are concatenations of proactive routes and depend only on the continuance of a cluster head, which can be predefined based on timers or some such criteria. Hence route failure probability is reduced considerably.

MMT Scalability: Clustering, hierarchical clustering and hybrid routing are used to address scalability. MMT-based algorithms have the above features. Besides, MMT algorithms do not flood messages, operate in a distributed fashion, and have very low overheads which enhance the scalability achieved through clustering and hybrid routing.

Robustness and Configurability of the MMT Algorithm: The cluster heads can be designated or elected in given area as they can have the same capabilities as the cluster clients and travel at same speeds and so on. The cluster sizes and the number of hops from a cluster head, can be defined, thereby providing a high flexibility in limiting the areas for proactive and reactive routing. Multi-tier clustering can be adopted if necessary. Amount of overlap across clusters can be defined and controlled. Amount of meshing in proactive routes within a cluster can also be defined and controlled. The MMT algorithm is simple and executed in a distributed fashion making it robust. MMT integrates into MAC layer to provide a comprehensive and compact protocol stack resulting in low energy, reduced overhead bits, lower processing complexity.

MAC protocols with MMT: The MAC protocols for the MMT-based framework were designed to leverage the VIDs used by the MMT algorithm and also to suit the applications requirements. Hence both random access and scheduled MAC protocols were used with the proposed solutions. Details are provided below

Design for a compact protocol stack with routing and MAC functions: The topic areas of major contribution relate to routing protocols, clustering algorithms and MAC protocols for mobile ad hoc networks. The significance in the proposed framework solution lies in the closely integrated operations of routing, clustering and medium access control as they all operate off of the meshed tree principle. To the best of our knowledge in the literature published thus far no solution targets an integrated clustering, MAC and routing solution to MANETs. Cross layered approaches, which break down the limitations of inter-layer communications to facilitate a more effective integration and coordination between protocol layers, is one approach that has similar goals. However, our solution is not a cross layered approach. We felt that in a dedicated and critical MANET application, such as the cognitive airborne networks, one should not be constrained by the protocol layers or stacks, but achieve the operations through efficient integration of required functions. The justifications for such an approach will be clear after we discuss the results.

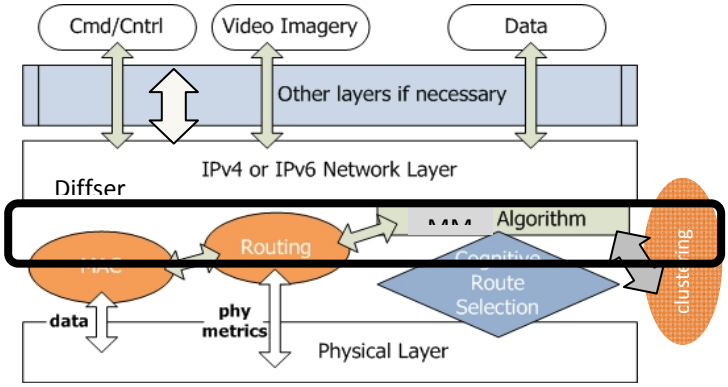

Figure 8 Integrated MMT-based Solution

The compact protocol stack was designed as shown in Figure 8. The combined clustering, routing, and MAC functions operate at layer 2. This is possible because clustering uses the meshed tree algorithm to form clusters around an elected cluster head which is also the root of the meshed tree. The tree branches provide the proactive routes to forward data between cluster clients and cluster head and vice versa. The creation of several overlapped clusters allows for scalability and robust connectivity. The MAC operations are based on the Virtual IDs used to build the meshed trees. Thus all functions related to clustering, routing and MAC are handled by a single algorithm. The solution operates at layer 2. However the presence of IP at layer 3 will not impact the operation of the layer solution. Thus the solution is transparent to protocols at layer 3. That is - applications can be run directly on top of the proposed framework or can be run over IP which could be the layer 3 on the proposed solution. In the first case applications can directly provide their service requirements to the meshed tree algorithm to decide on service specific optimal routes for the application. These routes can also be physical layer sensitive as the meshed tree algorithm operates at layer 2 and has direct access to physical layer parameters. In the sections that follow we provide the details of the different components that form the integrated solution.

3 Progress against Planned Objectives

Planned Objectives for the said period:

1. Extending the proposed Multi-Meshed Tree (MMT) algorithm to the wider AN role as shown in Fig 1. This involves mobility and vehicular velocity support and legacy integration
2. Physical layer integration involving radial velocity and cluster membership, power and energy awareness in route selection and antenna factors in clustering
3. Application layer integration to consider service specific handling
4. Evaluation involving merging MMT Opnet with AFRL's Phy Sim, prototype feasibility, and protocol refinement through simulations

The proposed work involved the following activities towards the planned objectives:

1. Design for a compact protocol stack with routing and MAC functions and gateways capable of addressing integration across heterogeneous networks;
2. An evaluation of the proposed design and its capability to handle quality sensitive application such as video images, command and control data and voice;
3. Studies on the interface to physical layer modeling and simulation capabilities available at Air Force Research Labs Rome, NY to more accurately assess the impact of physical layer metrics such as Doppler shift, MACH speeds of some AN nodes and variable link quality among others;
4. An evaluation of the solution's capability to incorporate physical and application layer metrics into the intelligent decision making, to make the AN node cognitive;

Details of results and technical achievements are provided under Section 4.

4 Results and Discussions

The proposed solution was investigated for heterogeneous tactical networks as encountered in airborne networks

1. Surveillance networks with omni-directional antennas
2. Peer-to-peer communications networks for ground troops
3. Peer-to-peer airborne backbone networks with omni-directional antennas
4. Surveillance networks with directional antennas up to one hundred nodes
    i. Comparing two link assignment
    ii. Optimizing link assignment based on MMT parameters
5. Peer-to-peer airborne backbone networks with directional antennas
    i. Based on hybrid scheduler
    ii. Based on distributed scheduler

The details of the MMT-based design in each case and the results with performance discussions are provided below.

4.1 Surveillance Networks With Omni-Directional Antennas

Surveillance networks are an essential application of Airborne MANETs. Depending on the type of surveillance required, it may be beneficial in certain applications to use 'mobile' nodes to perform surveillance. One such requirement arises in tactical applications, where unmanned aerial vehicles (UAVs) are used for aerial survey. Such MANETs of UAVs face severe challenges to deliver surveillance data without loss of information to specific aggregation nodes. Depending on the time sensitivity of the captured data, the end-to-end packet and file delivery latency could also be critical metrics. Two major protocols from a networking perspective that can impact lossless and timely delivery are the MAC protocol and the routing protocol. Physical layer and transport layer protocols will certainly play a major role; however, we limit the scope of this work to MAC and routing layer protocols. Further, these types of surveillance networks require several UAV's to cover a wide area while the UAV's normally travel at speeds of 300 to 400 km/h. These features pose additional significant challenges to the design of MANET routing and MAC protocols as they now must be both scalable and resilient: being able to handle the frequent route breaks due to node mobility.

In surveillance applications, data collected at the aggregation nodes has to be analyzed for action, if necessary. This is done at remote centers that siphon the data from the data aggregation nodes via satellites or fast moving and high powered UAV's, which periodically travel by the aggregation nodes. The predominant traffic pattern in surveillance networks hence is converge-cast, where data travels from several nodes to a collector node. We leverage this feature in the proposed solution. We also integrate routing and MAC functions into a single protocol layer, which we call the framework, where both routing and MAC operations are achieved with a single 'proposed' address. The routing protocol uses the inherent path information contained in the addresses, while the MAC uses the same addresses for hop-by-hop packet forwarding.

Data aggregation or converge-cast types of traffic are best handled through multi-hop clustering, wherein a cluster head (CH) is the special type of node that aggregates the data. Thus, a clustering mechanism is included as well in our integrated routing and MAC framework. The 'meshing' of the tree branches in MMT allows one node to reside in multiple tree branches that originate from the root, namely the CH. The duration of residency on a branch depends on the movement pattern and speeds of the nodes. Thus, as nodes move, they may leave one or more branches and connect to new branches. Most importantly, even if a node loses one path to the CH, it likely remains connected to the CH via another branch and thus has an alternate path. The MMT clustering scheme also allows for the creation of several overlapped multi-hop clusters leading to the notion of multi-meshed trees where each meshed tree cluster is formed around a CH node. The overlap is achieved by allowing the branches of one meshed tree to further mesh with the branches in neighboring clusters. This provides connectivity to cluster clients moving across clusters. It also helps extend the coverage area of the surveillance network to address scalability.

It is important to understand the cluster formation in the clustering scheme under consideration and the routing capabilities within the cluster for data aggregation at the CH. The multi-hop clustering scheme and the cluster formation based on the 'meshed' tree algorithm are described with the aid of Figure 9. The dotted lines connect nodes that are in communication range with one another at the physical layer. The data aggregation node, or cluster head, is labeled 'CH'. Nodes A through G are the CCs. For simplicity in explanation, the meshed tree formation is restricted to nodes that are connected to the CH, by a maximum of three hops.

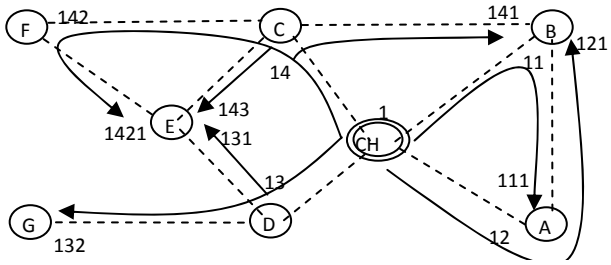

Figure 9 Cluster Formation-based on Meshed Trees

At each node several 'values or IDs' have been noted. These are the virtual IDs (VIDs) assigned to the node when it joins the cluster. In Figure 9, each arrow from the CH is a branch of connection to the CCs. Each branch is a sequence of VIDs that is assigned to CCs connecting at different points of the branch. The branch denoted by VIDs '14', '142' and '1421', connects nodes C (via VID '14'), F (via VID '142') and E (via VID '1421') respectively to the CH.

Assuming that the CH has a VID '1', the CCs in this cluster will have '1' as the first prefix in their VIDs. Any CC that attaches to a branch is assigned a VID, which will inherit its prefix from its parent node, followed by an integer, which indicates the child number under that parent. This pattern of inheriting the parent's VID will be clear if the reader follows through the branches identified in Figure 9 by the arrows.

The meshed tree cluster is formed in a distributed manner, where a node listens to its neighbor nodes advertising their VIDs, and decides to join any or all of the branches as noted in the advertised VIDs. Note that a VID contains information about number of hops from the CH. This is inherent in the VID length that can then be used by a node to decide the branch it would like to join if shortest hop count is a criterion. Once a node decides to join a branch it has to inform the CH. The CH then registers the node as its CC and confirms its admittance to the cluster and accordingly updates a VID table of its CCs. Thus the 'meshed tree' cluster formation allows a CH to control the nodes it accepts. A CH can restrict admittance of nodes that are within a certain number of hops or not admit new nodes to keep the number of CCs in the cluster under a certain value. This is useful to contain the data collection zone of a cluster.

Cluster formation and maintenance: The proposed meshed tree cluster formation in this work takes two parameters: the cluster size and the maximum hops at which to accept CCs. These two parameters are used by the CH during the cluster formation and its subsequent evolution as nodes move over time. Nodes in a surveillance network will announce themselves regularly using 'hello' messages. During the initial phase just after deployment of the UAVs,

only the CHs have a pre-assigned VID. The CHs will announce this VID in their 'hello' packets. Nodes within one hop will send a registration request to the CH. The CH will accordingly assign them a VID, which could be '11', '12' and so on until '19'. The assignment and joining is completed when a registration response packet is sent to the joining CC by the CH. Once a node has acquired a VID, it can then start advertising its own VID in a 'hello' message. This then allows nodes at two hops from the CH to join the cluster. Their VID is assigned to them by their parents, but the registration request will be forwarded to the CH, who decides if it can accept the CC and then generates a registration response to send to the CC. Thus the CH is able to control the dimensions of the cluster and the zone for data gathering. All CCs are expected to register with all of their CHs, and additionally update them when there are any changes in their VIDs.

Routing in the Cluster: The branches of the meshed tree provide the ***route*** by which to send and receive data and control packets between CCs and the CH. Consider packet routing where the CH has a packet to send to node E. As an example, the CH may decide to use the path given by VID '1421' to E. The CH will include its VID '1' as the source address and E's VID '1421', as the destination address and broadcast the packet. The nodes that will perform the hop-by-hop forwarding are nodes C and F. This is so, as from the source VID and destination VID, C will know that it is the next hop en route, because it has a VID 14 and the packet came from VID '1' and is destined to '1421' i.e. it uses a path vector concept. When C broadcasts the packet subsequently, F will receive and eventually forward to E. The VID of a node thus provides a virtual path vector from the CH to itself. Note that the CH could have also used VIDs '143' or '131' for node E, in which case the path taken by the packet would have been CH-C-E or CH-D-E respectively. Thus between the CH and node E there are multiple routes as identified by the multiple VIDs. The concept of support for multiple routes through multiple VIDs allows for robust and ***dynamic route adaptability*** to topology changes in the cluster. Nodes can request for new VIDs and join different branches as their neighbors change. This keeps the routes updated. The amount of meshing in the cluster of Figure 1 has been kept low, for picture clarity purposes. The meshing is limited only by the number of VIDs a node is allowed to acquire, and the maximum hops at which it can join the cluster, both of which can be set as tunable attributes for the clusters in the network.

Route failures: Capturing all data without loss is very important in surveillance networks used in tactical applications. Loss of data can be caused due to route failures or collisions at the MAC. There are two cases of route failures that can occur, yet be swiftly rectified, in the proposed solution. In the first case, a node may be in the process of sending data, and has even sent part of the data using a particular VID, only to discover that said VID or path is not valid anymore. In the second case, a node may be forwarding data for another node, but after collecting and forwarding a few data packets, this forwarding node also loses the VID which was being used.

Case I: Source node loses a route: For example, node B in Figure 10 is sending a 1 MB file to the CH using its shortest VID '11'. Assume that node B was able to send 0.5 MB, at which time due to its mobility it lost its VID '11' but was still able to continue with VID '121'. Node B can then send the remaining 0.5 MB of data using VID '121'.

Case II: Intermediate node loses a route: Let us continue the above example. Node A is forwarding the data from node B on its VID '12' (data comes from node B via its VID '121'). After sending 0.25 MB assume that node A moves in the direction of node D, loses its VID '12' but gains a new VID '131' as it joins the branch under node D. Node A can continue sending the rest of the file using its new VID '131'. As the knowledge about the destination node is consistent (i.e. it is the CH with VID '1'), any node is able to forward the collected data towards the CH thus reducing the probability of packet of data loss.

Disconnects: In a disconnect situation, a missing VID link may first be noticed by the parent or child of a node with whom the link is shared. In such cases, the parent node will inform the CH of the missing child VID, such that the CH will not send any messages to it. Meanwhile the child node, which is downstream on the branch, will notify its children about their lost VIDs (VIDs derived from the missing VID) so that they will invalidate those VIDs and not use them to send data to the CH.

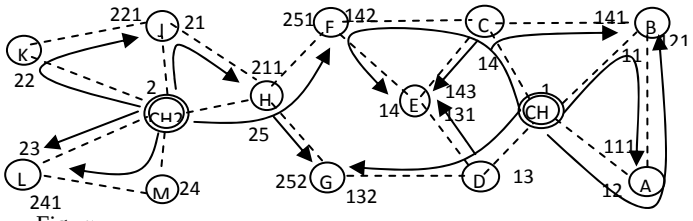

Figure **Figure 10 Overlapped Clusters-based on MMT**

Inter-cluster overlap and scalability: As a surveillance network can have several tens of nodes, the solution proposed must be scalable. We assume that several data aggregation nodes, i.e., CHs, are uniformly distributed among the non-data aggregation nodes during deployment of the surveillance network. Meshed tree clusters can be formed around each of the CHs with nodes bordering two or more clusters allowed to join the branches originating from different CHs. When a node moves away from one cluster, it can still be connected to other clusters, and the surveillance data collected by that node will not be lost. Also, by allowing nodes to belong to multiple clusters, the single meshed tree cluster-based data collection can be extended to multiple overlapping meshed tree clusters that collect data from several nodes deployed over a wider area with a very low probability of losing any captured information, thus addressing the scalability requirement in surveillance networks.

Figure 10 shows two overlapped clusters and some border nodes that share multiple VIDs across the two clusters. The concept is extendable to several neighboring clusters. Nodes G and F have VIDs '142', '132' under CH1 and VIDs '251' and '252' under CH2, respectively. Note that a node is aware of the cluster under which it has a VID as the information is inherent in the VIDs it acquires, thus a node has some intelligence to decide which VIDs it would like to acquire, i.e., it can have several VIDs under one cluster or acquire VIDs that span several clusters.

Comparison with Other Schemes: Under the related work section we highlighted several routing schemes, and frameworks that combined different types of routing algorithms and cluster-based routing. From the meshed tree-based clustering and routing scheme described thus far, it should be clear that our scheme adopts a proactive routing approach, where the proactive routes between CCs and CH in a cluster are established as the meshed trees or clusters are formed around each CH. Thus using a single algorithm during the cluster joining process a node automatically acquires routes to the CH. There is flexibility in dimensioning the cluster in terms of CCs in a cluster and the maximum hops a CC is allowed from a CH. The tree formation is different from the spanning trees discussed in the literature, as a node is allowed to simultaneously reside in several branches, and thus allowing for dynamic adaptability to route changes as nodes move. This also enhances robustness in connectivity to the CH. This approach is ideal for data aggregation from the CCs to the CH, and is very suitable for MANETs with highly mobile nodes. At the same time the coverage can be extended to several tens of nodes as shown with our simulation studies. We know of no work discussed in the literature with such unique properties. Though multiple overlapped clusters have been discussed in the literature [47, 48], our scheme achieves this in a simple manner.

After explaining the details of the clustering and routing process, we now explain how the MAC leverages the VIDs for efficient data aggregation at the CH. At the end of the explanation of the MAC operation we will again provide a comparison of the integrated framework that includes all three components.

Burst Forwarding Medium Access Control Protocol: The Burst Forwarding Medium Access Control (BF-MAC) is primarily focused on reducing collisions while providing the capability of MAC *forwarding* of multiple data packets from one node to another node in the same cluster. Additionally, the MAC allows for sequential 'node' forwarding where all intermediate nodes forward a burst of packets one after another in a sequence between a source and destination node through multiple hops. These capabilities are created through careful creation of MAC data sessions, which encompass the time necessary to burst multiple packets across multiple hops. For non-data control packets, such as those from the routing and cluster formation process, the MAC uses a system based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).

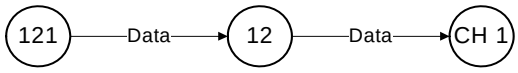

Figure 11 Traffic Forwarding-based on VIDs

Surveillance networks, being converge-cast, will result primarily in each node attempting to send data to a CH. As such, a node's data will physically travel up a VID branch to the CH in that tree. Therefore, by knowing which VID was used to send a data packet, and that packet's intended destination (the CH), an overhearing node can determine the next VID in the path. This process is used by all overhearing nodes to forward in their turn a packet all the way to the CH. This is illustrated in Figure 11, where when the node with VID 121 has data to send to CH1, the intermediate node with VID 12 will pick up and forward to the CH.

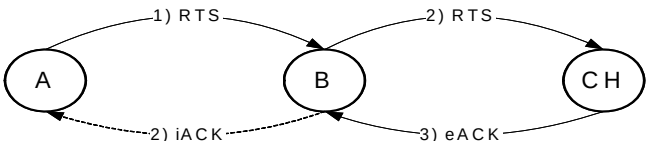

Figure 12 Setting a Data Session and Implicit Acks

The MAC process at a node that has data to send creates a MAC data session. A Request to Send (RTS) packet is sent by the node and is forwarded by the intermediate nodes till it reaches the CH. When a recipient node (i.e. a forwarding node) along the path receives the RTS, it becomes a part of the data session. A set of data packets may then be sent to the intended destination, in this case the CH, along the same path as the RTS packet. The final node in the path, the CH, will send an explicit acknowledgement (eACK) packet to the previous node for a reliability check as shown in Figure 12. eACKs are not forwarded back to the initial sender. Nodes in the path of the data session, except for the penultimate node, instead listen for the packet just sent to the next node. This packet will be the same packet being forwarded by the next node in the data session path (be it either an RTS or a data packet). Receiving this packet is an indication of an implicit acknowledgment (iACK), as the next node must have received the sent packet if it is now attempting to forward it. Not receiving any type of acknowledgment will cause a node to use the MAC retry model, discussed below.

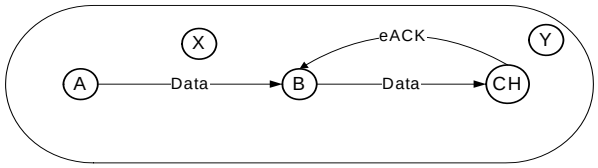

Figure 13 Use of NCTS Mode for Collision Avoidance

During a data session collisions from neighboring nodes are prevented in the same way as the collision avoidance mechanism in CSMA/CA. Nodes that hear a session in progress keep silent. When a node overhears an RTS, eACK, or data packet for which it is not the destination (or the next node in line to forward it), it will switch to a Not Clear to Send (NCTS) mode as shown in Figure 13. This will prevent a node from sending any control packets or joining a data session. If a node is already part of a separate data session, the node will continue with that data session. The NCTS mode lasts until the duration as specified in the Session on Wait (SOW) time. The SOW time is calculated by the initial sender within a data session, and marks the amount of time left for a particular data session. At each hop, it is decremented by the transmission time of the current packet plus a guard time to account for propagation delay as shown in Figure 6. When SOW time has elapsed, the data session is over and all nodes return to a Clear to Send (CTS) mode. A node in CTS mode may start a new data session, join a data session via forwarding, or send control packets.

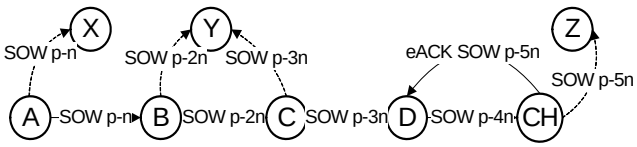

Figure 14 Session on Wait Time Calculations

Control packets from the routing and clustering process are queued and sent using CSMA/CA whenever a node is in CTS mode. To take further advantage of the MAC's data sessions in preventing possible collisions, nodes are also allowed to send control packets within a data session by extending SOW time a fixed amount.

Retry Model: The MAC stores any RTS or data packet sent into a retry queue. Until an eACK or iACK is heard for that packet, the packet will be retried up to three times within a single data session. Nodes will continue to receive data and issue eACKs for data packets while retrying the other packet, as shown in Figure 7. At the end of the data session, nodes will move any outstanding packets into their own data queues and will send them subsequently pretending to be the initial sender. If a packet fails to be sent in two separate data sessions, an error report is sent to the routing and clustering process for further action. This may require the use of a new route and thus is passed to the routing process.

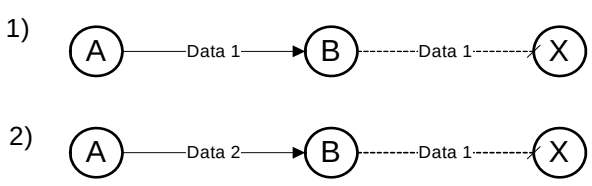

Figure 15 Data Retry Model

Comparison Stage II: It should be obvious to the reader by now that in the integrated framework the clustering process forms clusters based on meshed trees, where the tree branches are used for routing the packets from the CCs to the CH. The MAC brings the added capability of any node taking over and forwarding the packets to the destination and

uses the VIDs, which burst forward packets from CCs to the CH. It should also be clear to the reader, why we integrated the three functions into one layer. The primary reason is the natural operational dependency of all three schemes upon the one algorithm. Separating them into different layers would have resulted in suboptimal performance of the framework, which may not be an efficient solution for such critical applications as surveillance networks.

Simulation and Performance Analysis: We decided to conduct our comparison with two well known routing protocols OLSR and AODV. The first is a proactive routing protocol and the second is a reactive routing protocol. We use the proactive routing protocol OLSR to evaluate and compare with the performance of our solution to small networks with 20 nodes. Furthermore to make the studies comparable, we designated certain nodes as data collection nodes and the destination for data sending nodes in its vicinity. We used the reactive routing protocol to evaluate and compare the performances from the control overhead perspective in networks of sizes 50 and 100. In this case also the collection nodes were designated as destination for nodes in its vicinity. For completeness we evaluated OLSR, AODV and the MMT-based framework for all 20-, 50- and 100-node scenarios, with varying numbers of senders.

For OLSR and AODV we used the custom developed 802.11 CSMA/CA models available with Opnet at layer 2. These models provide flexibility in selecting optimal parameters and thus optimal operational conditions through proper setting of retry times, intervals for sending 'hello', 'topology control' or other control messages for OLSR and AODV. The scenario set up in the MMT solution in ns2 however faced constraints due to the random placement and selection of sending nodes as compared to selecting the nodes closest to the designated destination/CH as in Opnet. We therefore recorded the average hops between a source and destination node in all our test scenarios to serve as a baseline for comparison.

Simulation parameters: The transmission range was maintained at approximately 10 km. The data rate was set to 11 Mbps, the standard 802.11 data rates. No error correction was used for the transmitted packets and any packet with a single bit error was dropped. Circular trajectories with radii of 10 km were used (circular trajectories are used in lieu of elliptical trajectories as they introduce more stress into the test scenarios with more frequent route breaks). Some of the trajectories used clockwise movement, while some traveled counter-clockwise. This was done again to introduce stress in the test scenarios. The UAV speeds of the nodes varied between 300 and 400 km/h. 'Hello' interval was maintained at 10 seconds. The above scenario parameters were maintained consistent for all test scenarios.

The performance metrics targeted were

- Success rate, calculated as the number of packets delivered to the destination node successfully as a percentage of the number of packets that originated at the sender node.
- Average end-to-end packet delivery latency calculated in seconds.
- Overhead messages generated during data delivery; this was required as otherwise no overhead would be generated by the reactive routing algorithms unless there was data to send to some destination nodes. Overhead was calculated as the ratio of control bits to the sum of control and data bits during data delivery.

All the above performance metrics were recorded along with the average hops between sender and receiver nodes, for 20, 50 and 100 nodes, where the number of sending nodes was varied depending on the test scenario. The file sizes used for data sessions were each 1 MB and the packet sizes were 2 KB. In a session all senders would start sending the 1 MB file simultaneously towards the CH. We provide in-depth explanation for the 20 node graphs; the graphs in 50 and 100 nodes have a similar trend, hence we do not repeat the explanations.

Analysis of results for the 20-node test scenario: Figure 16 shows the four performance graphs based on results collected under the 20-node scenario. The number of senders was varied from 5 to 10 to 16, where in the last case as there were 4 data aggregation nodes, all other nodes, i.e., all CCs were sending data to their respective CHs. The average hops graph is provided as a baseline of comparison due to the difference in placement and selection of sending nodes in Opnet and ns2 as described earlier.

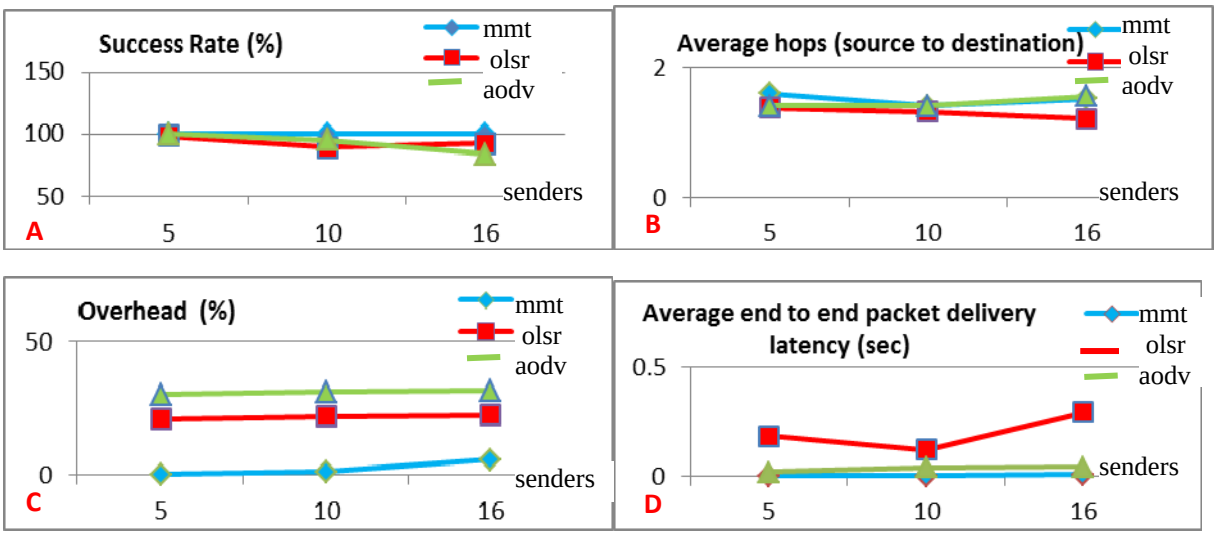

Figure 16 Performance Graphs – 20-Node Scenario (surveillance networks/omnidirectional)

Success rate and average hops: Graph A is the plot of the success rate versus the number of sending nodes. In the MMT-based framework, the success rate was 100% as the number of sending nodes was increased from 5, 10 to 16. For AODV and OLSR the success rate was high with 5 senders but decreased with an increase in the number of senders. While the success rate for AODV drops to 82%, for OLSR it dropped only to 87%. The success rate for OLSR with 10 senders is less than with 16 senders. This discrepancy will be clear if we look at the average number of hops (graph B) between sending and receiving nodes: with 5 senders the average hops recorded was 1.38, for 10 senders it was 1.32 and for 16 senders it dropped down to 1.22. This happened because the 5 senders selected first were further away from the designated destination node. In the case of 10 senders the added 5 senders were now closer to the destination node but when the last 6 senders were included they were still closer to the destination node bringing down the average hops, and thus were able to increase the success rate in packet delivery. However between 5 senders and 10 senders, due to the increase in traffic in the network, the average hops dropped by 0.6, yet the success rate still experienced a decrease. A similar explanation holds for the MMT framework too, where the average hops with 10 senders is lower than with 16 senders; however this did not affect the success rate and all packets were delivered successfully.

Average packet latency: MMT and AODV show very low latency as compared to OLSR (graph C). Due to the reduced success rates in the case of AODV, fewer packets were delivered and thus there is a dip in the average latency for 10 sending nodes, as the amount of traffic due to data packets is less in the network and also the packets which were taking longer did not make it to the destination. OLSR shows a higher latency due to the control traffic which delays the data traffic.

Overhead: The MMT framework solution has very low overhead compared to OLSR and AODV in all 3 cases of 5, 10 and 16 senders (graph D). This is probably a very unique feature of this framework solution. The reason for this can be attributed to the local recovery of any link failures as handled by MMT as compared to OLSR which requires resending the updated link information, or in the case of AODV, which has to rediscover routes if the cached routes are stale. The second reason could be the reduced collision and better throughput due to the BF-MAC. A point worth noting is that though MMT adopts a proactive routing approach, its overhead is very much lower than the reactive routing used in AODV even with fewer number of sending nodes, i.e., 5 senders.

Validation of the Comparison Process: It may seem to the reader that there are several improved variations of OLSR and AODV that may have performed better than just OLSR and AODV. However, it should be noted that the proposed framework outperforms OLSR and AODV significantly in all performance aspects, especially for the type of surveillance applications considered in the work. This is despite the fact that the average number of hops encountered between the sending and receiving nodes in the MMT framework is higher than OLSR by a significant

amount in all 3 cases and comparable to AODV for the 10 and 16 senders. This, as explained earlier was due to the lack of control in node placement and sender selection in ns2-based simulations.

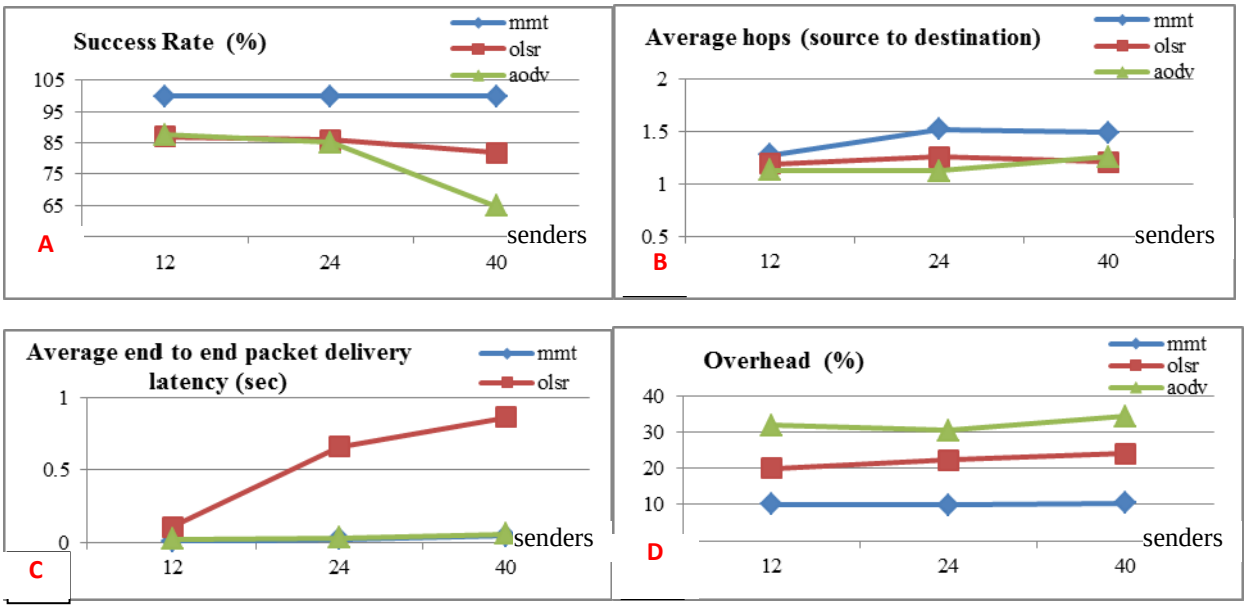

Figure 17 Performance Graphs – 50-Node Scenario (surveillance networks/omnidirectional

Analysis of results for the 50-node test scenario: Figure 17 shows the four graphs for the 50 node scenario. The MMT-based solution continues to maintain the success rate very close to 100% as the number of senders increased to 40, where all CCs send to their respective CHs. OLSR and AODV show a decrease in the success rate with AODV drop being higher than OLSR with 40 senders, which can be attributed to the increased number of senders, which is a well known phenomenon with reactive routing protocols. The average end-to-end packet delivery latency for OLSR is higher than AODV, because of the higher number of average hops with 20 senders and higher successful packets transmitted at 40 senders. The end-to-end packet delivery latency for MMT is still quite low and comparable to that achieved with AODV in which 15 to 35% of the packets were not delivered. The overhead with MMT is now at 10% compared with OLSR's around 20% and AODV with the highest overhead of over 30%.

Analysis of results for the 100-node test scenario: Figure 18 shows the four graphs for the 100 node scenario. While MMT consistently exhibits a similar performance as seen for the 20 and 50 nodes with a slight increase in the overhead and latency with increased number of senders with the average hops still greater than AODV and OLSR. OLSR shows a further drop in the success rate as compared to the 50-node scenario, which is due to the limitations faced when flooding the topology control messages. While the AODV success rate starts at 75% and drops to 68% for 40 senders and 47.5% for 80 senders, which is as expected. Overhead for AODV is higher than for the 50 nodes scenario as there are more discovery messages, while OLSR maintains the overhead between 20% to 30%.

Conclusion: The framework was especially designed to handle airborne surveillance networks for collection of surveillance data in a timely manner with the least data loss. We evaluated the framework and compared it with the two standard protocols, OLSR and AODV, using comparable network settings in each case. The performance of the proposed solution indicates its high suitability to such surveillance applications.

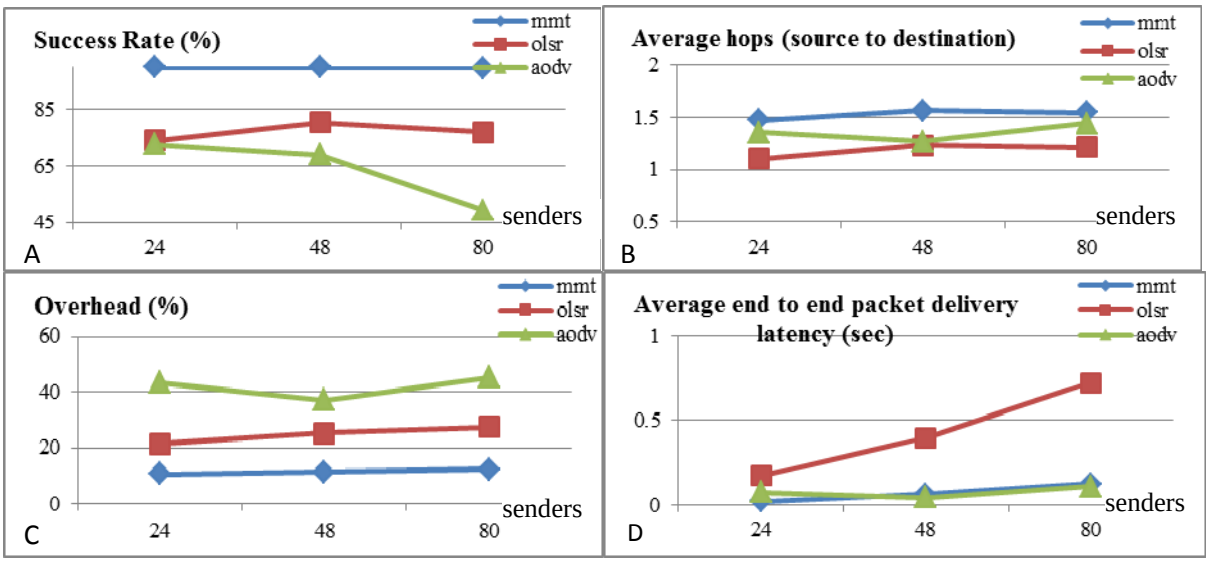

Figure 18 Performance Graphs – 100-Node Scenario (surveillance networks/omnidirectional)

4.2 Peer-to-Peer Communications Networks For Ground Troops

The MAC protocol was random access burst forwarding protocol as described earlier. However the simulation parameters were set using a random waypoint mobility model with 1-5 m/s speeds to account for pedestrian and slow moving vehicle speeds. The performance parameters were the same as for the last case. However the file sizes were reduced to 10 kbyte files. The picture below is captured from Opnet simulations to indicate the placement and movement of the nodes.

Analysis of 20 node scenario: The number of sending nodes are 4 to 11.96 in Figure 19. The reasons for the fraction in sending nodes was in some seeds not all nodes were able to send the files, due to heavy cross traffic in the network. Further work needs to be conducted to extend the tests to a greater number of sending nodes.

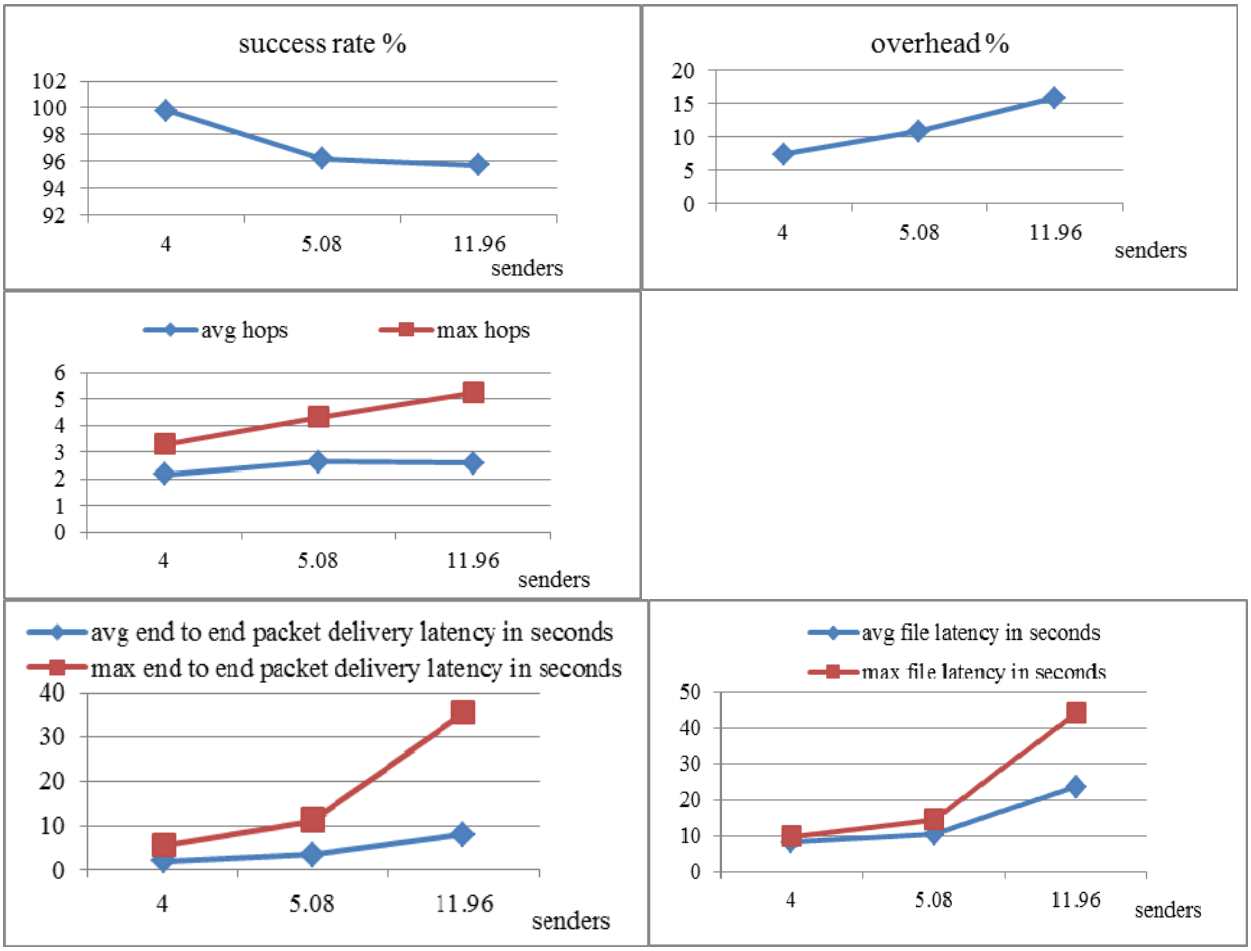

Figure 19 Performance Graphs – 20-Node Scenario (peer-to-peer/ground troops)

4.3 Peer-to-Peer Airborne Backbone Networks With Omni-Directional Antennas

The simulation set up and protocols were the same as for the last two cases. However the node mobility and speeds were maintained as for the surveillance network scenario. The results obtained for various test scenarios are provided below with explanations. The number of sending nodes was again varied as shown in the graphs. Some nodes did not start sending hence the number of sending nodes recorded is a fraction. This requires fixing and optimizing the solution.

Analysis of 20- and 50-node scenario: Similar to the results noticed for the 20-node scenario, some nodes were unable to start initiation of traffic. However of fifty nodes 27.6 were able to send traffic and the performance recorded indicates a high success rate. Similar test with other protocols are not available in the literature and hence a comparison is not possible.

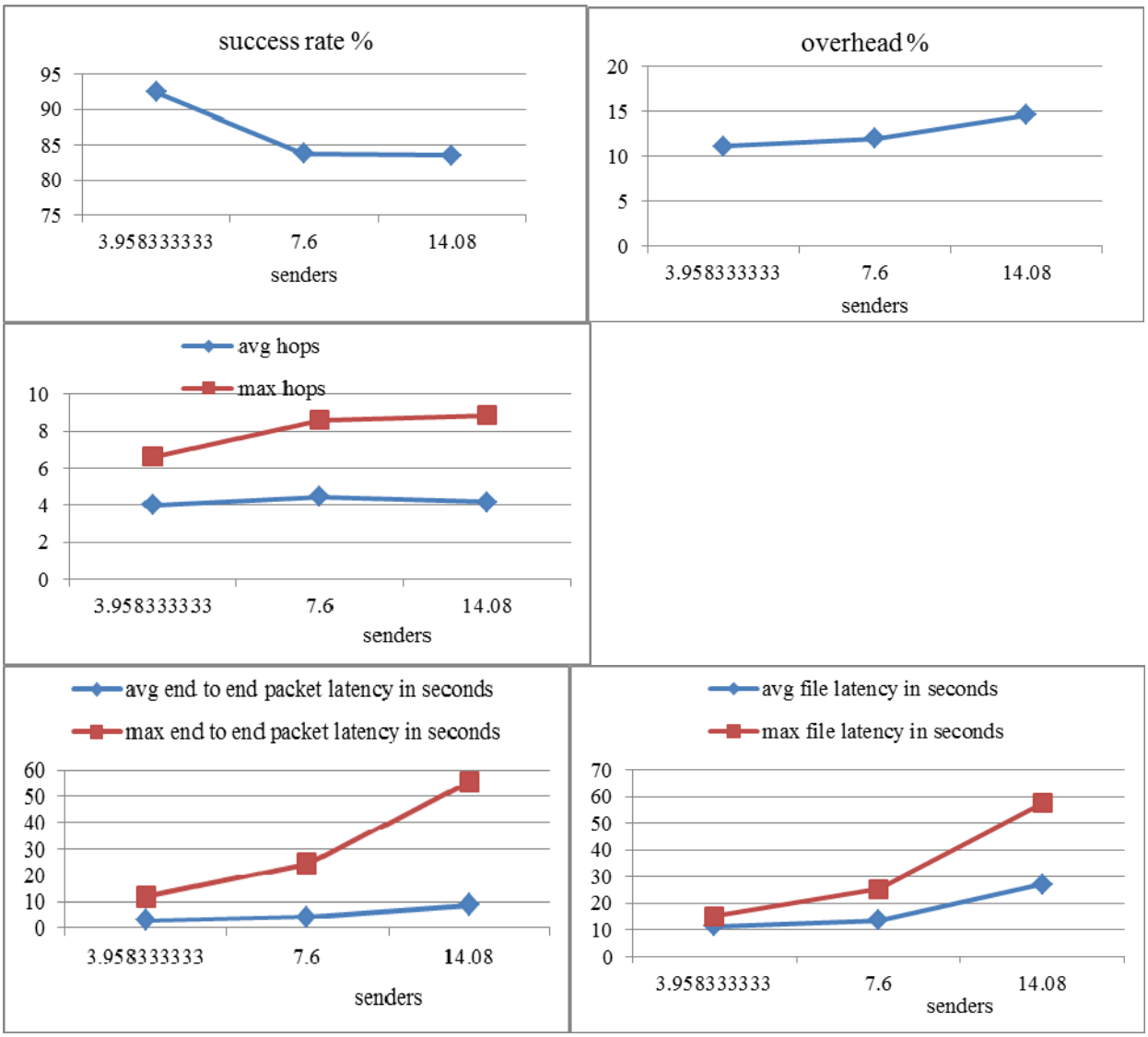

Figure 20 Performance Graphs – 20-Node Scenario (airborne backbone/omni-directional)

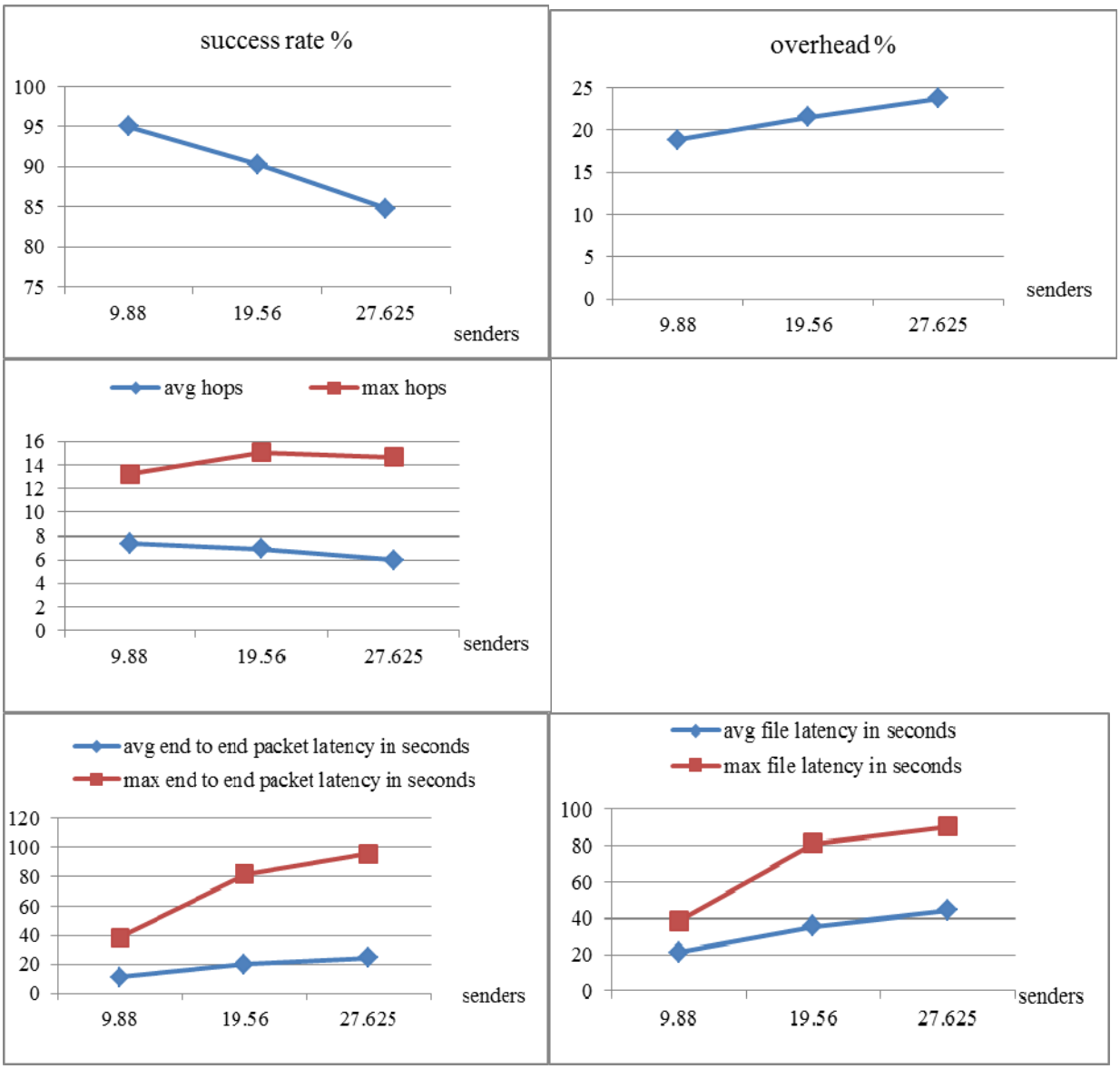

Figure 21 Performance Graphs – 50-Node Scenario (airborne backbone/omni-directional)

4.4 Surveillance Networks with Directional Antennas

4.4.1 Comparison of Two Link Optimization Strategies

While MANETs face several challenges in other protocol layers as well, this paper primarily addresses MAC and routing issues faced in a surveillance network of UAVs equipped with directional antennas. The main contribution of this work is the scheduling algorithm adopted in the MAC that uses time-division multiplexing for multiple access. The scheduling algorithm has the following features:

1. It is based on a multi-hop clustering scheme, and schedules time slots for nodes in the cluster to aggregate surveillance data at the CH.
2. It improves spatial reuse, due to directional antennas, which allows multiple transmissions in a single time slot.
3. It is aware of the routing mechanism within the cluster and hence schedules slots for data routing to the CHs in an efficient manner.

To address scalability to a large number of UAVs that could be deployed in a surveillance network, multiple overlapping clusters are used. Furthermore, high UAV speeds (in the range of 300-400 Km/h), which pose mobility challenges are addressed by the dynamic adaptability of the routing mechanism in the cluster.

To achieve higher capacity and better delay guarantees in networks, *Spatial*-reuse *Time-Division Multiple Access* (STDMA)-based MAC can be employed. In STDMA multiple transmissions can be scheduled as long as any interference at the receiving nodes does not impact its successful reception of packets [50]. In this manner, STDMA takes advantage of the spatial separation between nodes to reuse the time slots. Generally, such schemes require strict time synchronization among participating nodes for effective transmission and reception of packets. In addition, if the nodes are mobile, periodic changes in the network topology require updated STDMA schedules, with minimal computational complexity. It is also necessary that the updated schedule be propagated to all nodes concerned in a timely and efficient manner.

One of the most challenging tasks in any time-division multiplexing-based multiple-access scheme is generating schedules that manage network resources efficiently. Furthermore in STDMA, since multiple nodes can simultaneously transmit in the same time slot, an optimal scheduling algorithm must allow high reuse of time slots. Several algorithms have been proposed towards efficient scheduling in the literature [52-60]. In some algorithms, the scheduling function is performed by a centralized scheduler. This requires information about all nodes and their links in the network to be available to the centralized scheduler, which is a difficult task to achieve in a timely and resource-efficient manner. On the other hand, distributed scheduling can be done at the expense of higher complexity in the algorithm, i.e., one that can intelligently allow nodes to decide their schedules, such that there are no conflicts. In distributed scheduling, only nodes in the region of the change will act on it and update their schedules, without the collection of information at a central node.

Slot Assignment Strategies: The transmissions rights for a node during a time slot can be assigned using two different strategies. In a *link assignment* strategy, a node is allowed to communicate with a *specific* neighbor node in a time slot and in a *node assignment* strategy, a node is allowed to communicate with *any* of its neighbor nodes during its time slots. While node assignment strategy outperforms link assignment strategy at lower frame lengths, the latter is better under high traffic loads. As shown by Martinez and Altuna [52], the benefits of using directional antennas are greater when using link assignment in STDMA-based schedulers.

Our approach adopts hybrid scheduling, which is possible due to the cluster-based approach. Hence within a cluster a CH is the scheduler, making it a centralized approach. However each cluster has its scheduling done by its CH, which makes the approach distributed across the clusters and the solution is scalable. The proposed algorithm adopts link assignment strategy as link information for CCs in a cluster becomes available to the CH.

Scheduling features of the cluster: The meshed tree cluster is formed in a distributed manner, where a node listens to its neighbor nodes advertising their VIDs and decides to join any or all of the branches. Once a node decides to join a branch, it informs the CH, who registers the node as its CC and confirms its admittance to the cluster and accordingly updates a VID table of its CCs as shown in Table 1. Thus the 'meshed tree' cluster formation allows a CH to control the nodes it accepts, i.e., a CH can restrict admittance of nodes who are within a certain number of hops and not admit new nodes to keep the number of CCs in the cluster under a certain value. This is useful to contain the scheduling zone of the CH.

Table 1 VID table at the CH

Node	Multiple VIDs
A	12, 111
B	11, 121, 141
C	14
D	13
E	131, 143, 1421
F	142
G	132

Table 1 shows the VID information of CCs in the cluster (from Figure 9) maintained by the CH. Implicit topology information is available in this table; for example node B has a VID 1421, indicates that it has a link to the node

with VID 142. The CH will use this information and its capabilities of controlling and communicating with the CCs to establish recurring time frames with a time slot scheduled for transmission and reception on the links between CCs and between the CH and CCs in the cluster. As nodes join and leave a cluster, the CH updates this table and announces the new schedule. Thus the scheduling operations are closely integrated with the cluster formation process.

Inter-cluster scheduling: Given that a surveillance network can have several tens of nodes, the solution has to be scalable. Assuming that several data aggregation nodes, i.e., CHs are uniformly distributed among the non-data aggregation nodes during deployment of the surveillance network, meshed tree clusters can be formed around each one. Border nodes are allowed to join branches originating from different CHs and are expected to inform their respective CHs about their multiple VIDs under the different clusters; this information will enable the CHs to avoid conflicts when scheduling their time slots. When a node leaves a cluster it can remain connected to other clusters without losing the surveillance data collected by that node. Allowing nodes to belong to multiple clusters, single meshed tree cluster-based data collection can be extended to *multiple* overlapping *meshed tree* clusters that collect data from nodes deployed over a wider area with low probability of losing any captured data.

The Physical Layer: In this section we describe the operational features of the directional antenna system used at the physical layer. All nodes in the surveillance network are assumed to be equipped with four phased array antennas capable of forming two beamwidths: one focused with a beam angle of 10^o and the other defocused (angle 90^o); beam switching time is assumed to be less than 1.5 ms. In the focused beam mode the data rate is 50 Mbps and in the defocused mode the data rate is 1.5 Mbps. Each antenna array covers a quadrant and is independently steerable within that quadrant in the focused beam mode. Note that in reality, beams formed by directional antennas do not cover only a single quadrant; secondary (side) lobes may allow transmitted packets to be received outside the quadrant of interest. This is called the *burn-through* problem and is resolved by the MAC as explained in the next section.

Broadcast packets are transmitted using defocused beams on two diagonally opposite antenna arrays. When the cluster formation takes place during the initialization phase, all nodes except the CH have their antennas in defocused receiving mode.

We assume each node is equipped with Global Positioning System (GPS) to provide node position. Every node appends its GPS location to the packets it transmits. Receiving (neighbor) nodes log and continuously update a "location" cache with the transmitting node's location. Cached location information is used to track and estimate the current location of neighboring nodes during packet transmission. The cache stores a maximum of the last three positions of any node. The estimated location of the receiver node is computed by quadratic interpolation if all three positions are cached, by linear interpolation if only two locations are recorded, and by (default) the last known position if there is only one entry. The estimated location of a receiver node is used by a transmitting node to control the transmit power and form a directed beam to the receiver node.

GPS is also used for time synchronization. We assume that all nodes are synchronized to time slot boundaries and the beginnings of new frames. A guard time is included in each time slot to offset synchronization errors as well as allow for beam switching, e.g., for a 5 ms slot, a guard time of 1.5 ms was used. This was calculated to allow for a maximum switching time for antenna beams and also to accommodate high propagation delays when the transmitting and receiving nodes are far apart.

The Scheduling Algorithm: In this section we describe the STDMA-based scheduling algorithm. To explain the interaction between the STDMA scheduler, the MAC and the meshed tree clustering scheme we use Figure 22. The VID (link) information is passed to the scheduler by the MMT-based clustering module. Some parameters like the frame size (in terms of number of slots), time slot and guard time are provided as input to the scheduler. Node

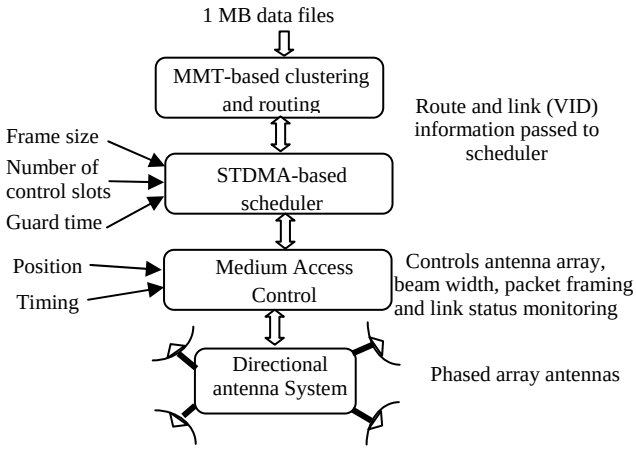

Figure 22 Scheduler Operation with Other Modules

position and timing information is collected by the MAC. The scheduler passes the schedule information to the MAC, which controls the antenna arrays to activate the most appropriate ones, to control the beamwidth and check for link status. Another function of the MAC is detection of *burn-through*, which is eliminated by checking sender and the next hop node's UID in the received packets. If the sender and receiver are the same, then the packet was received by the sender's antenna which was in receive mode. A neighbor node that overhears a packet will check the next hop ID and discard the packet if it was not the intended recipient.

The scheduling algorithm interacts with the MAC and the MMT cluster formation module and schedules time slots to support:

- Cluster formation after deployment of the UAV nodes.
- Cluster and route maintenance within the cluster after clusters have been formed.
- Dynamic accommodation for changes in network topology due to node movement and, if any, nodes joining.
- Time slot scheduling for data aggregation using link assignment strategy from CCs to CHs in each cluster.
- Periodic distribution of schedules to CCs in an efficient manner using established routes in each cluster.
- Scheduling for link maintenance packets at the MAC layer to check for link status.
- Early prevention and/or resolution of inter-cluster conflicts.

We used schedules with a frame length of 40 time slots each of 5 ms duration. We assume a node can either transmit or receive in its assigned time slot, but not both.

Control slots: Of the 40 time slots, 4 'control' slots are preselected for control purposes; it is assumed that all nodes are aware of the chosen control slots. These slots are used by nodes to advertise their VIDs, broadcast information, and listen to advertisements from neighboring nodes. Of the four control slots, two are 'transmit' slots (control-*tx*), and two are 'receive' slots (control-*rx*). The decision on which pair of control slots is control-*tx* and control-*rx* is decided as follows: To allow all nodes to interconnect effectively, it is necessary that all nodes receive advertisements sent by their neighbors. Nodes with even UIDs advertise in the first and third control slots using them as control-*tx* slots, while nodes with odd UIDs would use the same slots as control-*rx* slots and listen to advertisements. In the second and fourth slots, nodes with even UIDs would transmit, while the nodes with odd UIDs would receive. Packets transmitted in control-*tx* are broadcast packets, and transmitted using defocused beams on two diagonally opposite antenna arrays. In control-*rx* slots, nodes receive using all antennas in focused mode. Since each antenna covers a 90° quadrant, the antennas *scan* their respective quadrant using focused 10° beams in the control-*rx* slots. When a node has completely scanned its quadrant, it randomly selects which of the control slot pairs to use as control-*tx* and control-*rx.* This pattern of usage for the control slots is repeated.

Data slots: The remaining 36 slots are used for transferring data packets and other control packets between CCs and CH. From a node's perspective, assigned data slots can either be used for reception or transmission since nodes

cannot transmit and receive in the same slot. Transmission and reception in assigned data slots are done using focused beams. Unassigned data slots, i.e., slots not yet assigned by the CH, are announced by nodes during the cluster formation process and also when they would like to acquire new VIDs or update their VIDs. Unassigned data slots by default are used for reception in defocused beam mode. An unassigned slot is used for transmission if and only if the slot is also unassigned in the destination node. The use of unassigned data slots will be described as we proceed.

Cluster formation with directional antennas: During the initialization phase, following deployment of the UAVs, the non-aggregation nodes form clusters around the CHs. Thus at the start of this phase all data slots are considered unassigned by all nodes and all have their antennas in receiving mode with defocused beams. The cluster formation starts with nodes that have VIDs announcing their GPS location, VIDs, and their unassigned slots in a 'hello' advertisement packet. Note, immediately following deployment of the nodes, as per our assumption, only the CHs have a pre-assigned VID and hence are the only nodes transmitting.

Joining of one-hop nodes: The CH transmits a 'hello' packet using a control-*tx* slot. Upon reception of the 'hello' packet, the first hop node, say node B in Figure 9, sends a Registration Request (RREQ) packet to the CH using an unassigned time slot which it has in common with the CH – in this case it could be any of the 36 data slots. The RREQ includes all of B's unassigned slots – here, all 36 slots. The CH receives the RREQ packet, accepts node B and assigns it a VID which is attached to a Registration Reply (RREP) packet destined for node B. The RREP packet is queued by the CH until the end of the frame. At the start of the next frame, the CH transmits the RREP packet to node B using an unassigned data slot in common with an unassigned data slot of B. This delays cluster schedule creation for the consequent frame, allowing the CH to receive any further RREQ packets from other neighboring one-hop nodes. In this manner, cluster schedules are updated only at the end of frames to accommodate changes in the network due to newly joining nodes. More importantly, to avoid synchronization errors due to propagation delays, the schedule determines transmission rights for the subsequent frame. Figure 23 illustrates this sequence of receiving requests, schedule creation, schedule announcement and adoption in a cluster. Note that the sequence shown is repeated for every frame, i.e., while in frame *n*, the schedule is sent and new RREQs are received by the CH which are then processed at the end of frame *n*.

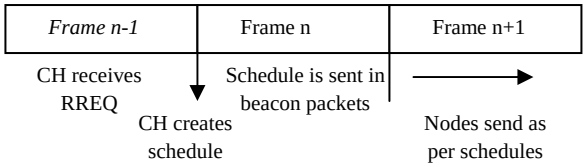

Figure 23 Operational Sequence in Scheduling

Joining of two-hop nodes: Once node B has acquired VID '11', it can then advertise its 'hello' packet in its control-*tx* slots. Assuming that node A has acquired one-hop VID '12' under the same cluster, node A now hears the 'hello' packet from node B and decides that it wants to join branch '11' and sends a RREQ packet to node B using an unassigned slot of node B which coincides with one of its own unassigned slots. Node B receives the RREQ packet, including the VID it has assigned to node A in the RREQ packet and forwards it to the CH on its assigned data slot along with any data that it has for the CH. As stated, the CH will queue RREP packet for A until the end of the frame when it assigns slots for A. In the next frame, the CH sends the RREP packet with the schedule for A to B using its assigned slot with B. B will forward the RREP for A using a common unassigned slot.

Data Slot Allocation: When node B joins the cluster, the CH assigns time slot 1 for transmission from CH to node B as shown in Figure 24 and time slot 40 for transmission from node B to CH. When the CH accepts node A, it will then assign slot 2 for transmission from CH to node A and slot 39 (the mirrored slot) for transmission from node A to CH. This scheduling operation is repeated at the end of each frame. All data packets are to be acknowledged. As noticed between any pair of nodes there is a slot for transmission in either direction. Hence when node B sends packets to the CH during its transmit time slot, the CH will acknowledge the packets during the next transmission time slot that the CH has for sending data to node B.

Link Assignment Strategies: The scheduling algorithm described above assigns a time slot for each link (VID) existing between a pair of nodes. For instance, node A can have multiple slots to node B i.e. via its VIDs '12' and '111', where with VID '12' A is the parent node and with VID '111' A is the child node. The algorithm will thus have 2 slots for transmission from A to B and 2 slots for transmission from B to A. This strategy can result in lower latencies for traffic between certain pairs of nodes and is dependent on the VID requests and allocation in the cluster – we call this the ***'Link assignment 1'*** strategy. The slot reuse becomes more complex, however, since there are redundant slot allocations which depend on the VID redundancy. Another issue that may arise is in the case where the number of links based on the VIDs is higher than the 36 slots that were set aside for data. Hence we decided to also investigate a link assignment scheme where between a pair of nodes only one data slot is assigned for transmission and one data slot for reception – we call this the ***'Link assignment 2'*** strategy. The second scheme requires the scheduler to base its decisions not only on the VIDs, which gives the link information, but also use the UIDs of the nodes to restrict the data slots to one pair of transmit and receive slots between any 2 communicating nodes.

Link maintenance: The MAC is capable of detecting link loss first as it occurs when a node moves out of communication range of another node. The detection is based on loss of response from the neighbor node. To detect link loss, the MAC at a node generates and sends a "link maintenance" packet in every assigned slot scheduled for transmission, even if there are no other packets to be sent by that node. Upon reception of a packet, the receiver node is aware that the link to that transmitting node is active. Link loss is assumed when a node does not receive any packets in an assigned data slot for 3 consecutive frames. This information is then sent to the MMT layer as notification to release the related VIDs.

Conflict Avoidance/Resolution: In some situations, a CC bordering 2 or more clusters can be assigned the same time slot by its CHs for communication with its neighbor nodes. Border nodes avoid this by including their slot assignment in a Registration Update (REGU) which they send to all their CHs upon joining a new cluster. The old CHs update the border CC's schedule using this information when reallocating slots. Conflicts can still occur due to propagation and routing delays, however, when sending REGU packets. To prevent this, CHs include the current size of the cluster in RREP packets. Hence when a conflict occurs in a border CC's schedule, the border CC adopts the schedule from the larger cluster and sends a "conflict packet" to the CH of the smaller cluster requesting reassignment of the conflicting slots.

Another conflict that can occur is when a parent node is also a border node. The parent node has to check the scheduled time slots for its children with its CHs for any schedule conflicts. If a conflict is detected, a "conflict avoidance" packet is sent to that specific CH to reassign conflicting slots.

Location Updates: The location information in the cache maintained by the CCs is restricted to neighboring nodes. The CH can use the location information of all its CCs to schedule time slots for multiple transmissions between pairs of nodes such that interference is minimized. Hence each CC sends its cached location information in 'location' packets to all their CHs. Such location packets are transmitted only on assigned slots using focused beam patterns.

Schedule announcement: The cluster schedule is distributed by the CH to all nodes in the cluster at the start of each frame in "beacon" packets. Beacon packets are transmitted only on assigned slots using focused beam patterns. Each node independently chooses the 'best' (in our case shortest, which is decided on the VID length) route to forward the beacon packet using MMT's routing information. Transmitting schedules one frame ahead also allows enough time for the beacon packets to reach nodes that are at the maximum hop limit from the CH.

Sample Schedule: In this subsection, we present a sample schedule based on *Link assignment 1 strategy* for the cluster in Figure 9. The slot allocation is given in Figure 24. Each column is a slot; we show only 12 slots, which is a partial frame. In the first column are the node UIDs, which in this case are the alphabets we used for the CCs in Figure 9. In each column we mark the VID of the sending and the receiving nodes with the arrows showing the direction of transmission, for example, in slot 1 CH VID '1' sends to node A VID '11'. The slot allocation process proceeds by allocating slots from the CH to one-hop nodes, followed by the one-hop CCs sending to their two-hop children and the two-hop CCs sending to their three-hop children and so on. However, due to the directional antennas used, we can have simultaneous transmission between two pairs of distinct nodes, for example, in slot 3 CH is sending to node D on VID '13', but node B using VID '11' is sending to node A at VID '111'. A closer look at the latter half of the schedule will reveal that the flow from the outer leaf nodes to the CH is the mirror of the allocation process from CH to leaf nodes, i.e., the 1st hop children are allocated the last time slots in the frame, thus allowing for data aggregation as the packets are moved towards the CH.

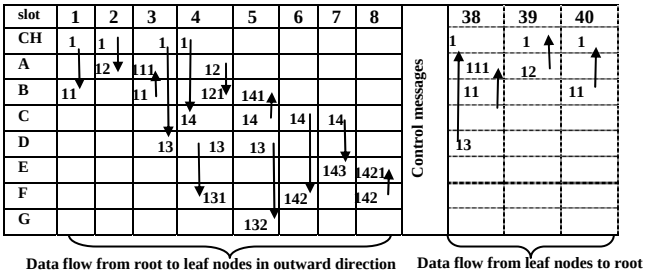

Figure 24 Sample schedule for Cluster in Figure 9

Simulation Results: We performed simulations using Opnet for 20-, 50- and 100-node multi-hop networks. Nodes in the networks were randomly placed in an area of 400km x 400km for the 20-node networks and 1600km x 1600km for the 50- and 100-node networks. All nodes were randomly assigned clockwise and counter-clockwise circular trajectories, with 100 km radius and speed of 300 km/h at 5 km altitude. The phased array antennas were modeled using Opnet's directional antenna model. We modified the radio pipeline stages in Opnet with the simulation parameters displayed in Table 2.

Table 2 Opnet Simulation Parameters

Simulation Parameters at Physical Layer	
Propagation Model	Free Space
Bandwidth	300 MHz
Center Frequency	15 GHz
Data rate	Defocused Beam Pattern – 50 Mbps, Focused Beam Pattern – 1.5 Mbps
Beam Angle	10° (for Focused Beam Pattern)
Minimum SINR	14 dB
Bit Error	Based on QPSK Modulation Curve
Maximum radio range	350 km (default)
Transmitting Power	0.1 – 0.9 W
Antenna Gain	Defocused Beam Pattern – 6 dB, Focused Beam Pattern – 23 dB

Targeted performance metrics included:

- success rate, calculated as the number of packets delivered to the destination node successfully as a percentage of the number of packets that originated at the sender node,
- average end-to-end packet delivery latency calculated in seconds,
- average end-to-end file delivery latency calculated in seconds, and
- overhead message generated during data delivery, calculated as the ratio of control bits to the sum of control and data bits during data delivery

Nodes in each network were randomly selected to send 1 MB files simultaneously to the closest CH in 2 kB packets. Overhead, average hops, packet delivery rate, as well as mean packet and file latencies were measured. Each simulation was run with 40 different seeds and the average values were plotted in the graphs shown in Figures 25-27. In all graphs the parameters were determined for the two types of link assignment strategies that we adopted. The top-left graph in Figure 25 is the plot of the successful packets delivered to the CH versus the number of CCs that sent the packets. The number of CCs sending 1 MB file was varied from 5 to 10 to 16, where in the last case all CCs in the 20-node scenario sent their data to the 4 CHs. The success rate is around 100%, dropping slightly for the link assignment 2 strategy at 16 senders. In terms of the average hops, there is essentially no difference between the two strategies as noted in the top-right graph. The overhead gives an idea of the amount of control messages used by the proposed system as a percentage of the data traffic and is given in the bottom-right graph. Depending on the way the overhead is calculated, when the number of data sending node increases the overhead may be reduced. The

mean packet delivery latencies and file delivery latencies are plotted in the bottom-left graph. The decrease in latencies could be attributed to decreasing average hops.

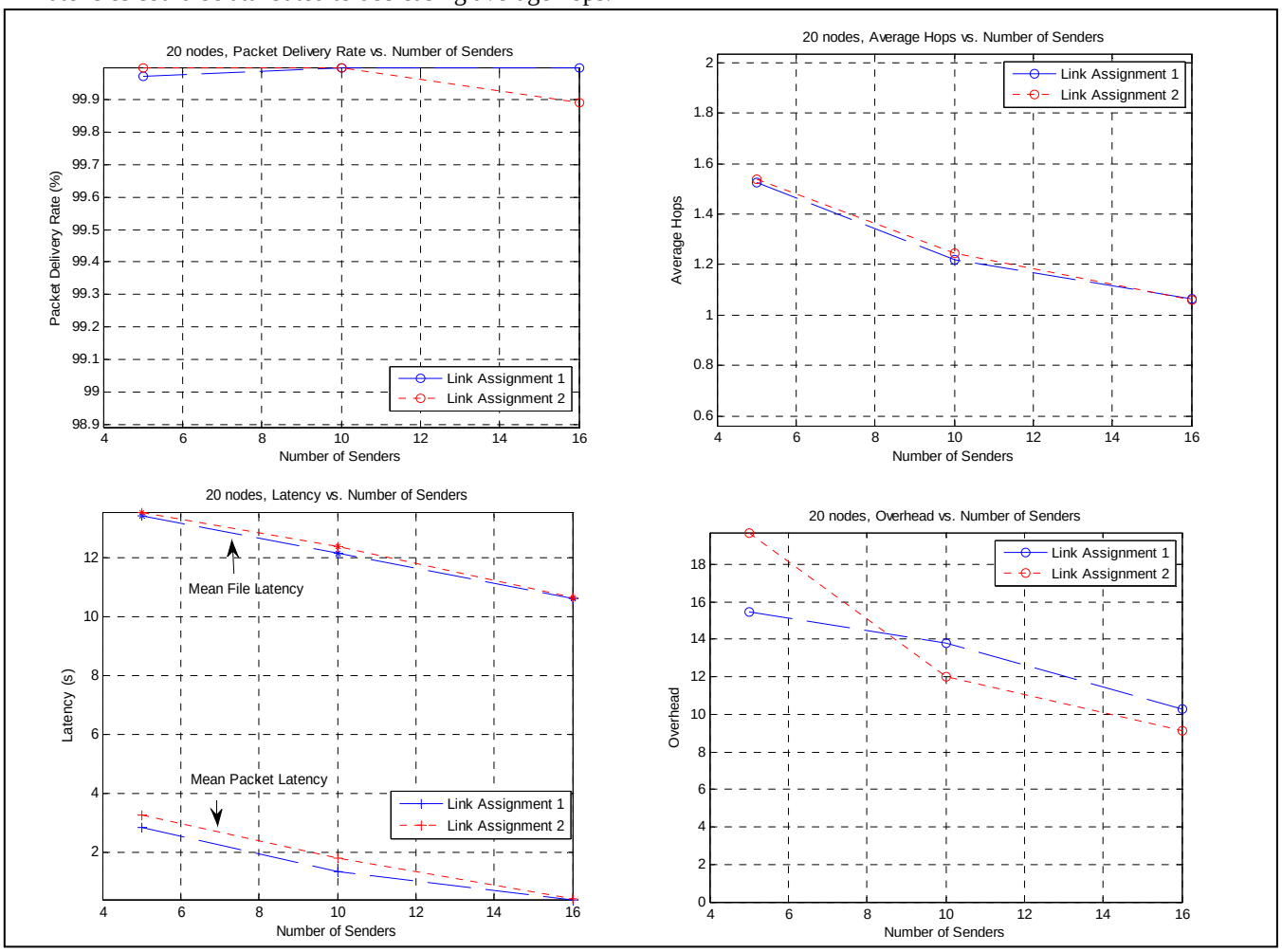

Figure 25 Performance Graphs – 20-Node Scenario (surveillance networks/link assignment strategies)

Figure 26 has the 4 graphs for the 50-node scenario. As the number of sending nodes varies from 10 to 24 to 40 nodes, where in the last case all CCs were sending a 1 MB file to the 10 CHs, the success rate in packet delivery drops slightly to 99.85 when all CCs send data to their CHs. The average hops in this case is around 1.25, which is different from the 20-node scenario. This is due to the way in which the sending nodes are selected. For example in the graph for 20 nodes, the 5 sending nodes would have been at a higher number of hops and as the other nodes were included the average hops decreased as new nodes were closer to the CH. However in the 50-node scenario the random selection resulted in more uniform placement of the sending nodes as the number of senders increased. The variation in average hops, from that of the 20-node scenario, may also be due to the larger network size which produced a flattening effect in average hops. The mean packet latency increased gradually in the 50-node scenario, a result of increased traffic and packet buffering. A similar trend is noticed for the file latencies as well with a mean file latency around 12 seconds. Again the variation between link assignment strategy 1 and 2 is insignificant.

Figure 27 provides the four graphs for the 100-node scenario. The success rates are again around 99.9 % as the sending nodes varied from 20 to 48 to 80, where in the last case all CCs are sending to the 20 CHs. The average hops has a trend similar to that noted for the 50-node scenario. Mean packet latencies and average packet latencies are also similar to the 50-node scenario.

The overheads in all three scenarios are relatively low, i.e., less than 20%. However, one notices that the overhead for Link assignment 2 strategy is less than the overhead for Link assignment 1 strategy due to the fact that under Link assignment 2 strategy the slots are more sparsely occupied as we allocate only one slot per pair of nodes for transmission and one slot for reception; this is in contrast to Link assignment 1 strategy, where multiple slots are assigned between the same pair. This results in less conflict under the Link assignment 2 strategy.

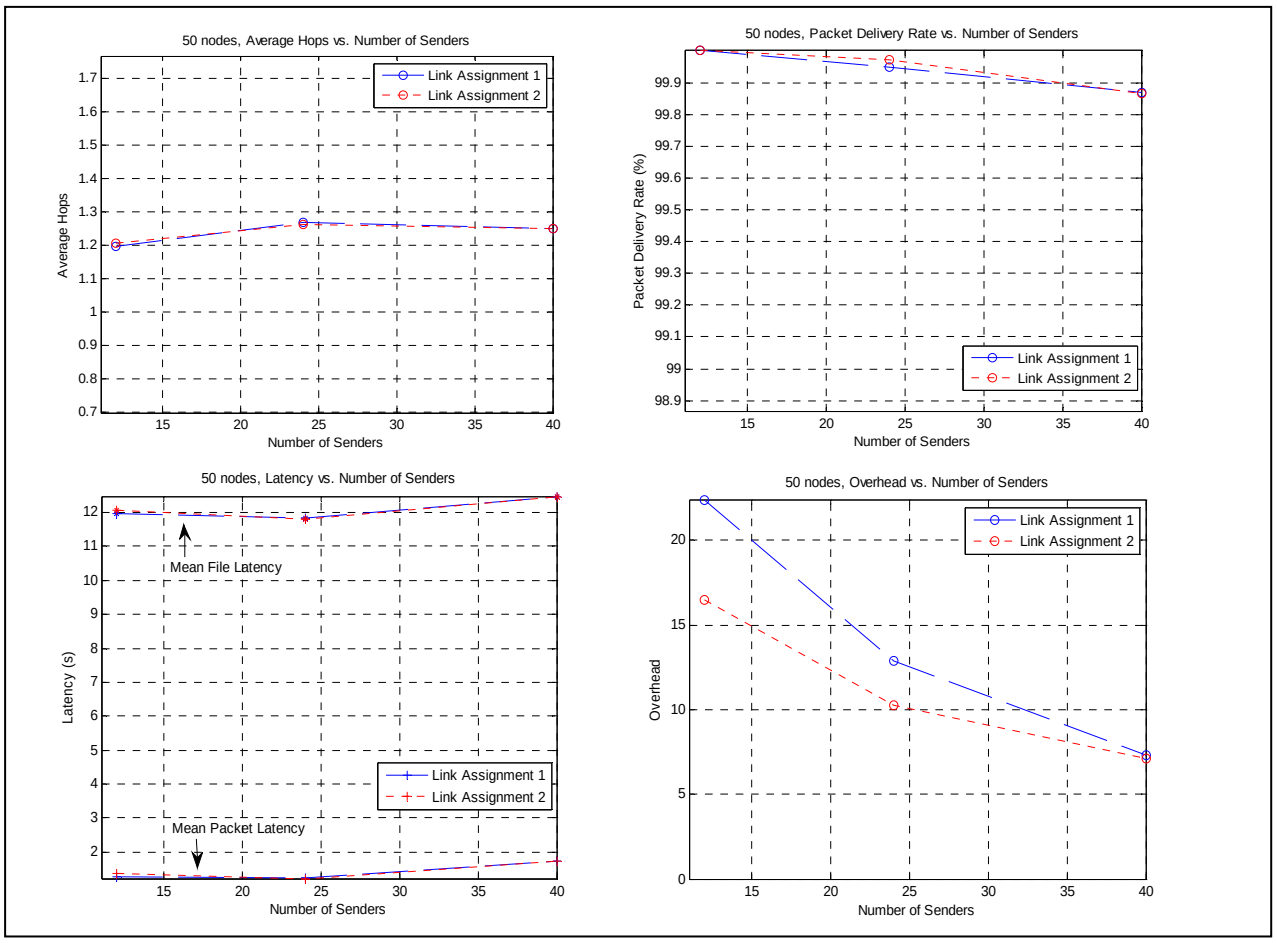

Figure 26 Performance Graphs – 50-Node Scenario (surveillance networks/link assignment strategies)

A point worth investigating is the insignificance of the Link assignment strategies in all performance graphs and for the 3 different network scenarios of 20, 50 and 100 nodes, except for the overhead, where consistently Link assignment 2 strategy has less overhead than Link assignment 1 strategy. This is why we did not observe lower latencies with Link assignment 1strategy, where each node was getting relatively more slots to send the traffic. Higher overhead as stated in the last paragraph is due to more conflicts with Link assignment 1 strategy; this could backlog and buffer the traffic at the nodes, resulting in higher latency. Moreover, under Link assignment 1 strategy, although all nodes had the probability of getting higher number of slots in a frame, for some nodes the slot allocation would have been postponed for frames due to the high slot occupancy due to multiple slots being assigned per link. This requires investigation of the slot use and is one of the points for future improvements as noted earlier.

These results indicate that despite a scheduling algorithm which was not optimized, the performance of the system as a whole for data aggregation type applications, as required in surveillance networks, is very effective. The focus during the system development was to achieve a high success rate in packet delivery and the performance indicates that we were able to achieve this. Furthermore the proposed system is scalable to one hundred nodes that are moving at high speeds.

We investigated two link assignment strategies called the Link assignment 1 strategy and Link assignment 2 strategy. We provided comparative results between the two strategies for surveillance networks of 20, 50 and 100 nodes. The performance results indicate the high packet delivery ratio that the proposed algorithm is able to achieve even under stressful conditions where all nodes in the network were sending a 1 MB file to their respective CHs. The overheads are low and mean packet delivery latencies are less than 2 seconds if the average hops of the sending nodes are around 1.25. There are several aspects of the proposed algorithm that need further investigation and fine tuning. Nevertheless, the preliminary results presented are very promising.

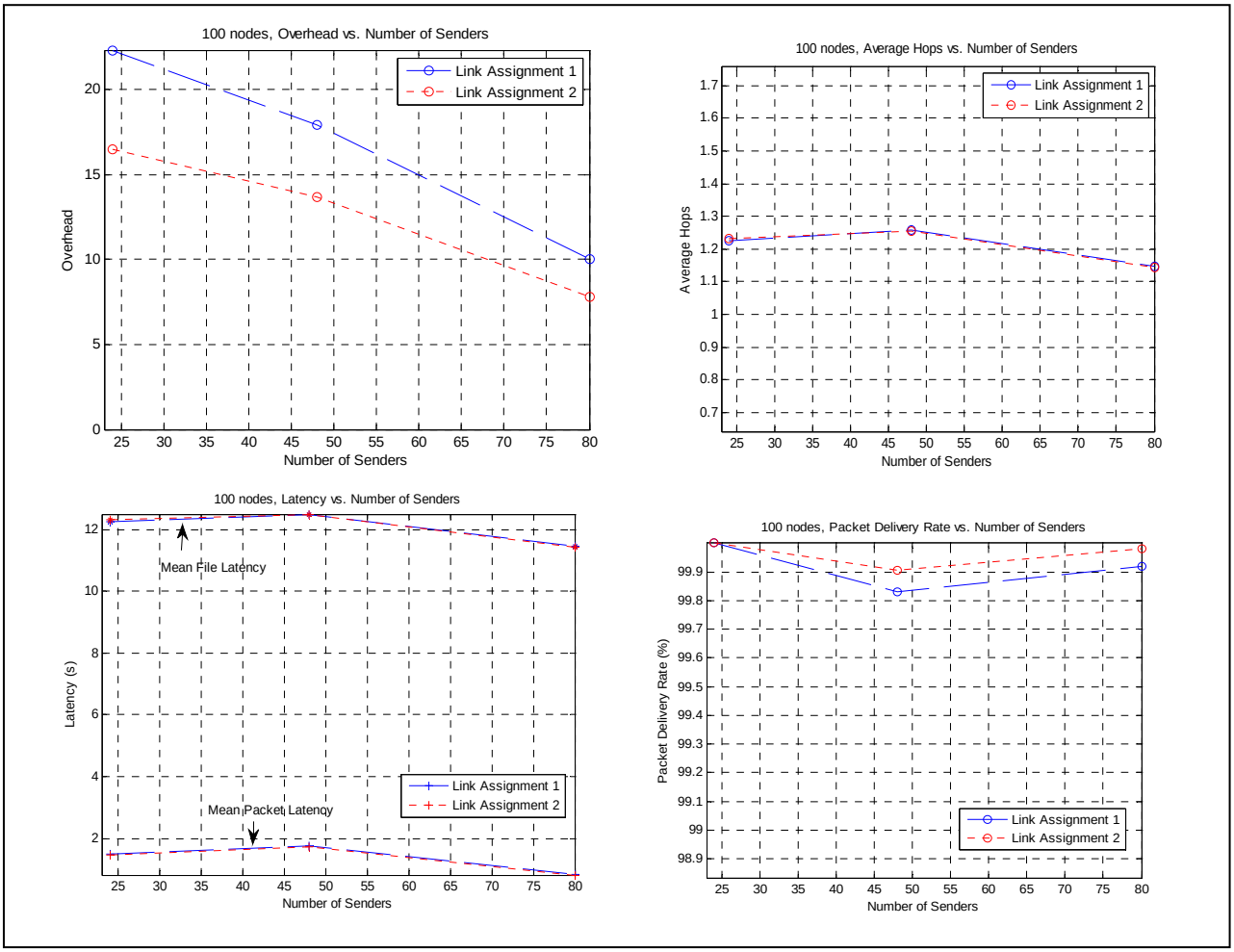

Figure 27 Performance Graphs – 100-Node Scenario (surveillance networks/link assignment strategies)

4.4.2 Optimizing Link Assignment to MMT Cluster Parameters

We will now present the possible optimization options under two categories: (1) MMT-based cluster attributes and (2) slot and frame timings.

Optimization based on MMT clusters: The MMT-based clustering allows optimizations of the scheduling strategy by tuning the cluster size, the maximum hops that a CH can support from a CC, the maximum number of VIDs allowed to each CC and the interval for sending 'hello' messages; the last parameter helps maintain network connectivity and thus improves the overall performance of the scheduling algorithm indirectly.

Let us take the example of the impact of cluster size on scheduling. In MMT-based clustering the cluster size determines the number of links that are visible in the cluster and that have to be scheduled by the CH. The number of recorded links in the cluster depends on the number of VIDs allowed to each CC, the maximum number of hops

allowed to each CC, and the communications range of the beams and the node density. Given that our current MMT clustering algorithm allows VID assignment on a first-come, first-serve basis, the VIDs assigned to a CC may not be optimal in terms of the number of hops from a CH. In the case of multiple meshed trees that are used to form the multiple clusters, a node is allowed to opt for a VID under the different clusters. We further allow a node to have only one outstanding VID request at any time, meaning that the VIDs can be requested only in a sequential manner. In this report, we focus on using some of the cluster attributes to assign time slots for link usage in an optimal manner within the given constraints.

The graphs in Figure 28 are provided to indicate the impact of cluster size and maximum number of VIDs, with the maximum number of hops set at four hops from the CH. To keep the graph simple, we present only three cluster sizes and three sets of multiple VIDs, i.e., 4 VIDs, 5 VIDs and 6 VIDs per CC. Frame sizes were set to 0.2 seconds, guard times to 1 ms with a total of 16 time slots of 12.5 ms each. The maximum hops from the CH was restricted to four in all cases. With 20 nodes and 4 clusters, the senders were set to 16 nodes which is the maximum number of data collection nodes in the surveillance network. We notice that with a cluster size of 12 the performance in terms of success rate is better than for a cluster size of 8 and 10. This is because a larger cluster size allows more nodes and expands the number of possible selections of VIDs in terms of hops acquired. For a cluster size of 8, we noticed poor performance attributed to reduced connectivity available to nodes.

Figure 28 A shows the plot for the average number of slots used by the scheduler. As expected the slots used for cluster size 8 is less than that for cluster sizes 10 and 12. The lack of a linear relationship is attributed to the complex relationship of the cluster parameters and their behavior for a given test scenario. For the typical test surveillance scenarios that we plan to use for evaluating the link assignment strategy, a cluster size of 12 with five VIDs would be an effective set of parameters to use.

Impact of Slot and Frame Timings: Depending on the application, the time slots can be determined such that they enhance the performance of the application. For example, if the traffic is predominantly data, then using large time slots and sending as many packets in a time slot will result in better performance, especially with highly mobile nodes that incorporate routes that are transient. However, if the application were of a transaction type or for voice

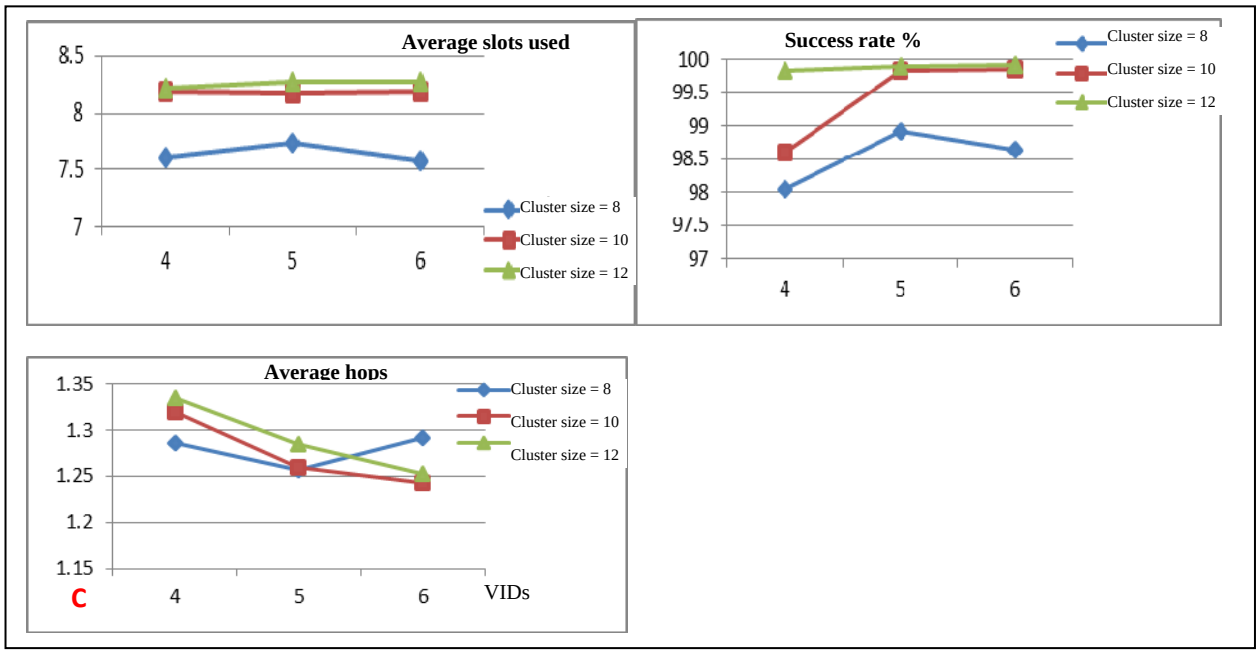

Figure 28 Performance with Varying Cluster Sizes (surveillance networks/directional)

traffic, maintaining a time slot that encapsulates a single transaction packet or smaller voice packets would result in better performance.

Obviously in surveillance applications, especially with highly mobile nodes, a relatively longer time slot would be efficient. However given that we use uniform time slots for data and control slots, irrespectively, we may be wasting

time during the control slots and also wasting the time slots allocated to nodes that have no traffic to send. The second factor is important especially in the synchronous transmission mode adopted in the TDM scheme; the ratio of unused slots can be high when there are few senders. We could have used all the slots for the few senders. Also, in the example of surveillance networks, slots for handling the converge-cast traffic heading towards a CH can be larger than the slots predominantly carrying non-converge-cast traffic away from the CH. Furthermore, typically for a given time slot, we would like to have a frame size that is able to handle all cluster clients in the cluster. Based on our link assignment strategy, this frame size, in terms of the number of slots, reaches an optimal value based on the MMT cluster properties.

Constant Time Slot with Varying Duration: Figure 29 depicts link assignment strategy performance in a 20-node scenario with 16 senders and 5 ms time slots as the number of time slots is varied. The optimal cluster sizes and VIDs from the previous studies were used. However, there is a marginal (less than 1%) decrease in the success rate, which can be attributed to the increased interval in sending 'hello' messages and link maintenance used by the MAC to continually check the link status. The average and maximum number of slots used shows a stagnant effect, as the scheduler for the given cluster attributes can use only a certain number of slots.

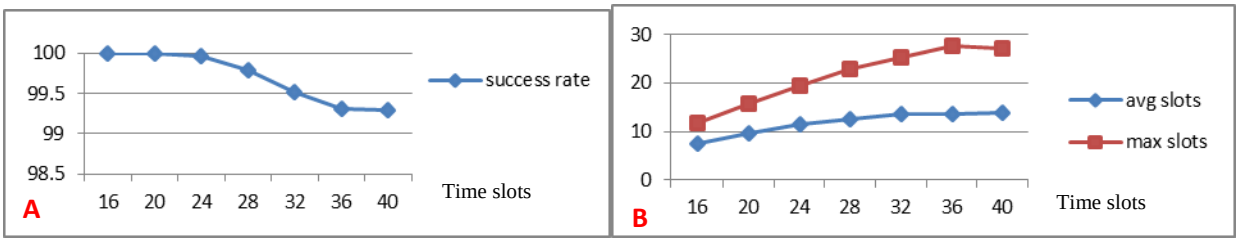

Figure 29 Performance to Varying Number of Time Slots (surveillance networks/directional)

Varying Time Slots with Constant Frame Sizes: As the number of slots is increased the time slot durations must necessarily decrease. For less than 16 time slots we notice that some nodes were not able to send data, resulting in a lower success rate. When using only 4 slots, only 2 nodes (one-hop) were able to send their data from the 16 enabled senders. The two nodes thus used 2 slots for sending and 2 slots for receiving with the remaining slots kept aside for control purposes. Graph B in Figure 30 has a predictive performance, where the actual number of used slots increases monotonically and then plateaus.

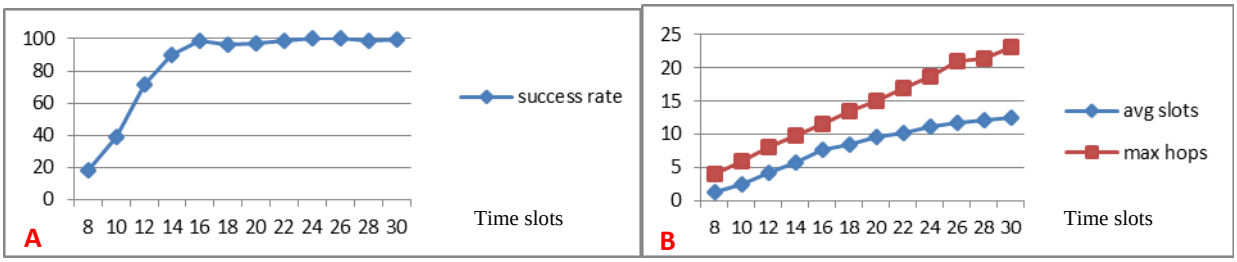

Figure 30 Performance with Varying Number of Time Slots (Fixed frame size)

Performance Analysis: All the above studies were conducted using the Opnet simulation tool. In this section we present the comprehensive results for the link assignment strategy that was optimized in terms of cluster size, allocated VIDs to cluster clients, and number of time slots. Tests were conducted for 20-, 50- and 70-node multi-hop networks. All nodes were randomly assigned clockwise and counter-clockwise circular trajectories, with a 10 km radius and a speed of 300 km/h at an altitude of 5 km. Maximum transmission range was restricted to 15 km and 5 nodes were allowed to reside in each trajectory; neighboring trajectories overlap. The phased array antennas were modeled using Opnet's directional antenna model.

Targeted performance metrics included:

- success rate, calculated as the number of packets delivered to the destination node successfully as a percentage of the number of packets that originated at the sender node,
- average end-to-end packet delivery latency calculated in seconds,
- average end-to-end file delivery latency calculated in seconds, and
- overhead message generated during data delivery, calculated as the ratio of control bits to the sum of control and data bits during data delivery

Other simulation settings included maintaining a data rate with the defocused beam of 50 Mbps and a data rate with the focused beam of 1.5 Mbps, with a beam angle of 10° for the focused beam. Nodes in each network were randomly selected to send a 1 MB file simultaneously in 2 kB packets to the closest CH. We measured overhead, average hops, packet delivery rate and mean packet and file latencies as stated above. Each simulation was run with several different seeds and the average values were plotted in the graphs shown in Figures 31-33. Our results highlight the effectiveness of the proposed scheme in achieving the targeted performance for surveillance networks.

20-Node Scenario: Figure 31 graph A shows the success rate, which is close to 100% and drops only slightly with 16 senders. In this case, all CCs are simultaneously sending 1 MB files to their CHs. The chances of schedule conflicts and node failures during the conflict resolution can be attributed to the slight drop in success rates.

The overhead plot shows an increase with 10 senders as compared to 5 or 16. As per our overhead calculations, this should decrease with more data traffic in the network. The increase in average hops with 10 sending nodes, as seen in Figure 31 Graph C, indicates that cluster formations produced more nodes at higher hops, accounting for the increase in overhead. With 16 senders the average hops of the CCs remained the same. Increasing senders and data traffic results in reduced overhead. In Figure 31, graphs D and E show that with increased senders in the network increased data traffic, increased backlogged traffic and increased packet and file latencies are reasonably expected.

A quick calculation shows that sending a single 2kB packet requires 0.3 ms. Hence in a 12.5 ms time slot, we can send around 40 packets. This would require 13 frames to send a 1 MB file, which would incur a delay of 3.25 seconds. This approximate calculation did not include the impacts of backlogs and multiple hops. If these factors were also taken into account then the maximum latency of 6 seconds recorded for packet latency would be acceptable.

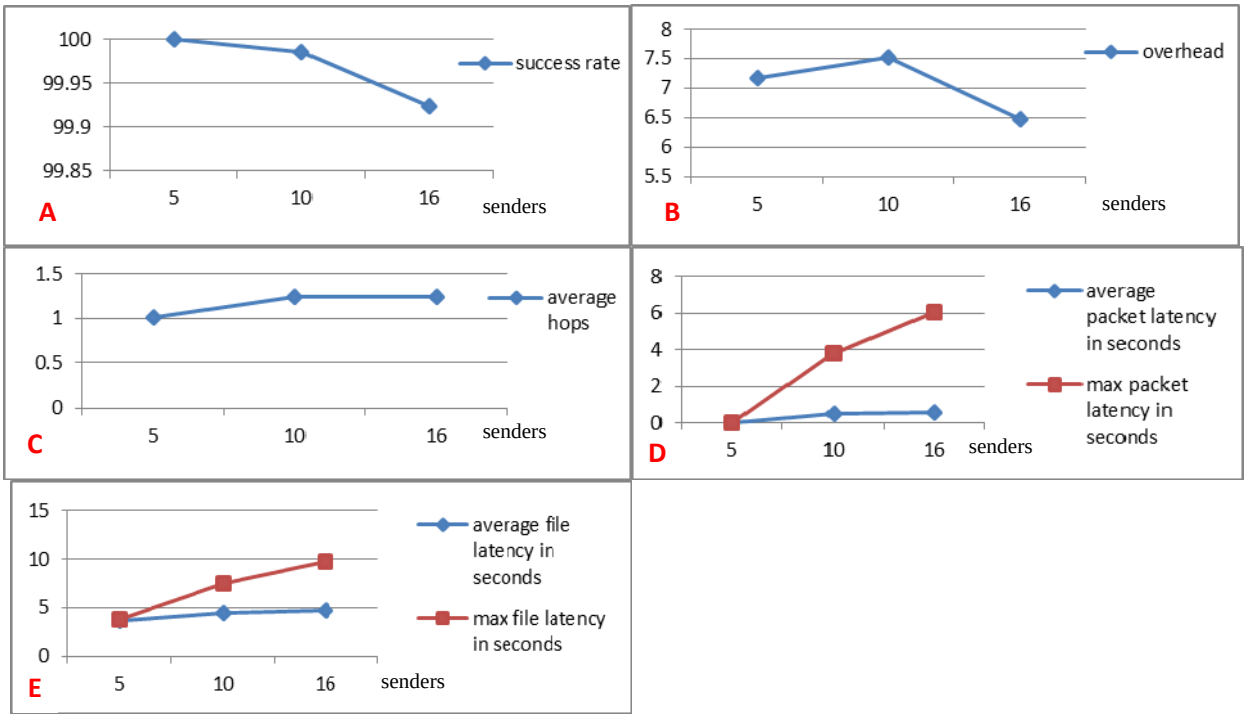

Figure 31 Performance graphs – 20-Node Scenario (surveillance networks/directional)

50-Node Scenario: In the case of the 50-node scenario, there were 10 clusters placed closed to one another such that there is a significant overlap and, hence, increase in the number of nodes facing conflict. All nodes that are non-peripheral have a greater probability of facing schedule conflict, which may result in their losing packets as well as their turn to send their data files; the latter having a higher impact with more senders. Another reason for packet loss may be outdated schedules, as every schedule is sent one frame ahead, which is 0.25 seconds, and a node may lose the link or VID in this interval. The success rate with 24 and 40 senders shows a significant drop to around 95%. The overhead and average hops trend similar to the 20-node scenario. As expected, the average and maximum packet latencies increase with an increased number of senders. We did not include the graph for file latencies as they are of no additional value.

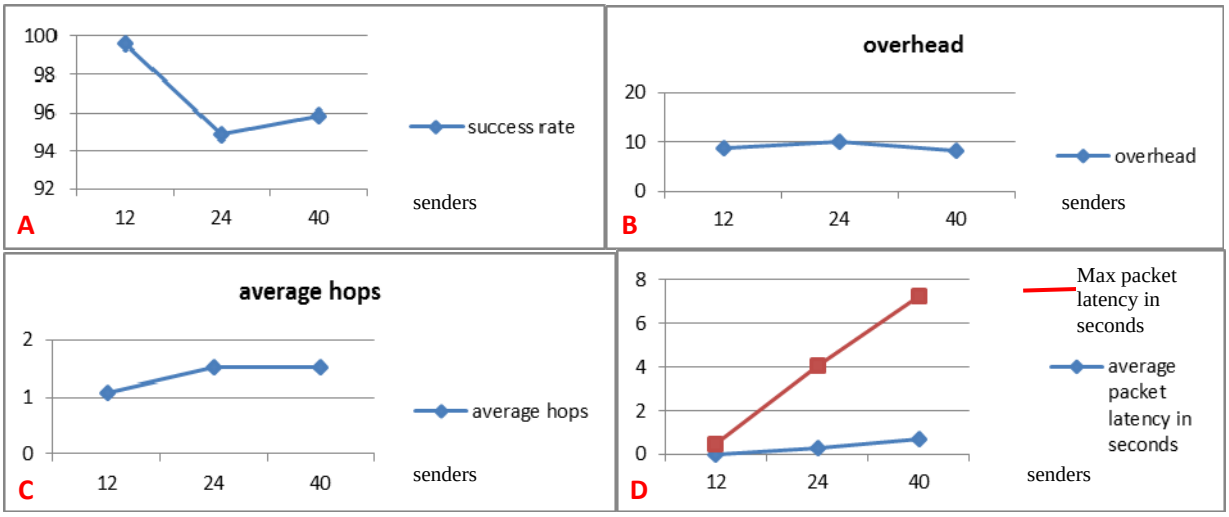

Figure 32 Performance graphs – 50-Node Scenario (surveillance networks/directional)

70-Node Scenario: For the 70 node scenario, there were 14 clusters. The success rate has dropped down to less than 86% with varying number of senders. However the reductions in success rate as the number of senders were increased is quite low, approximately 1.5 % as the number of sending users was increased from 14 to 56. In the case of 56 senders, again all CCs were sending 1 MB files to their respective CHs simultaneously. The overhead decreases with an increasing number of senders as the greater amount of data traffic in the network causes the ratio on control packets to data packets to decrease. The trends in packet delivery and file delivery latencies were similar to the 20 node scenario. We provide the packet delivery latencies graph D in Figure 33.

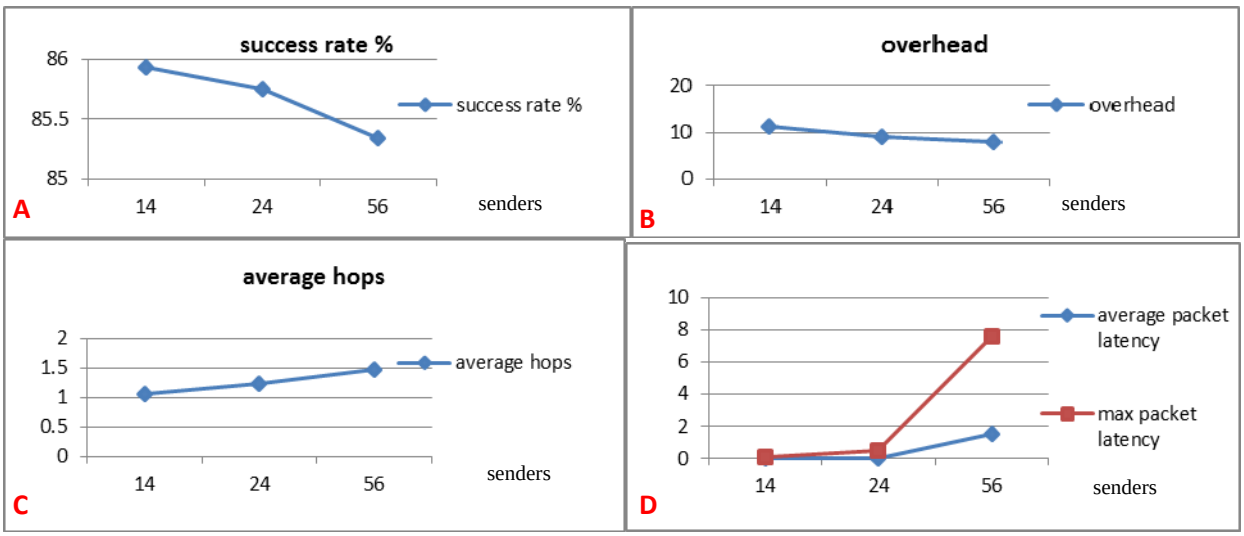

Figure 33 Performance Graphs – 70-Node Scenario (surveillance networks/directional)

4.5 Peer-to-Peer Airborne Backbone Networks with Directional Antennas

4.5.1 Hybrid Scheduler and Link Assignment Based on MMT

Backbone networks formed by airborne nodes such as *unmanned aerial vehicles* (UAVs) are of significant importance in tactical applications as they can be used to connect several tactical sub-networks which are at distance from one another. Though the targeted application is that of a backbone network, we limit our contribution to demonstrating the capability of the airborne backbone network to forward data reliably between two distant nodes capable of serving as gateway points to two distant sub-networks.

MAC and routing in MANETs (such as the airborne backbone network) present signficant challenges, especially when there are large numbers of highly mobile nodes in theater. While MANETs face several challenges in other protocol layers as well, this work primarily addresses MAC and routing issues faced in backbone airborne networks comprised of nodes equipped with directional antennas. The main contributions of this work are the reactive cluster-based routing protocol between pairs of distant nodes and its interworking with a TDMA scheduler adopted by the MAC protocol. Both the reactive routing protocol and TDMA scheduler base their operations on a 'meshed tree' clustering scheme.

The reactive routing protocol features:

- Reactive routes are a concatenation of proactive routes within the cluster. As the proactive routes are continually updated with node mobility, the probability of stale reactive routes is low.
- Reactive routes are maintained as a sequence of clusters. Combined with the above feature the probability of reactive route failures is low.
- Reactive route discovery and maintenance is done at cluster level. This reduces control message flooding.

The scheduling algorithm:

- Uses attributes of a multi-hop clustering scheme to schedule time slots for the CCs within the cluster.
- Is aware of the routing mechanism within the cluster and, hence, schedules slots for data routing from CH to CCs and from CCs to CH in an efficient manner.

Time-Division Multiple-Access Scheduler: TDMA scheduling requires strict time synchronization among participating nodes. If the nodes are mobile, periodic changes in the network topology will require updated TDM schedules to be computed, preferably with low complexity, and propagated to all nodes affected in a timely and efficient manner.

Scheduling algorithms fall under two main categories: centralized or distributed. In the centralized approach, scheduling is performed by a scheduler that requires information about all nodes and their links; this is a difficult task to perform in a timely and resource efficient manner, especially with large numbers of mobile nodes. Distributed scheduling requires complex algorithms with intelligence to enable each node to decide its schedule with minimal conflicts.

Our cluster-based approach allows us to use hybrid scheduling. Within a cluster the CH is the scheduler that decides the transmission reception schedules for its CCs. However each cluster's schedule is determined independently by its CH giving conflict consideration only to those CCs that are bordering two or more clusters thus making it distributed across clusters. Given that *link assignment strategies* are efficient if employed with directional antennas [13] and that the proposed clustering scheme has link information available, we employ the hybrid strategy.

Scheduling in the Cluster: The meshed tree cluster is formed in a distributed manner where a node listens to its neighbors advertise their VIDs and joins any or all of the branches. Once a node joins a branch, it informs the CH, who then registers the node as its CC, confirms its admittance to the cluster, and updates a VID table of its CCs.

From the CC's VID table 1, implicit topology information is available to the CH; for example, node B with VID '1421', indicates that it has a link to the node with VID '142'. The CH uses this information and its capabilities of controlling and communicating with the CCs to establish recurring time frames with a time slot scheduled for transmission and reception on the links between CCs and between CCs and the CH in the cluster. As nodes join and leave a cluster, the CH updates this table and announces the new schedule. Thus the scheduling operations are closely integrated with the cluster formation process.

Slots and Functions: A frame is comprised of control and data slots. In this work four slots are preselected for

control purposes. These slots are used by nodes to advertise their VIDs and other broadcast information as well as to listen to advertisements from neighbor nodes. Remaining slots are used for transferring data packets and other control packets between CCs and the CH. From a node's perspective, assigned data slots can be used either for reception or transmission using focused beams.

Slots not assigned as either control or data slots are considered temporary slots, which may be used to transmit packets when no slot has been assigned yet, such as during the registration process. When a node does not have anything to send on a temporary slot, it listens for any incoming transmissions. At any time these slots can be changed to an assigned data slot by the CH.

The cluster schedule is distributed by the CH to all CCs in the cluster at the start of a frame with 'beacon' packets. Each CC independently chooses the 'best' (in our case shortest, which is decided on the VID length) route to forward the beacon packet using the meshed tree's routing information. Schedules for a given frame are transmitted one frame ahead, allowing enough time for beacon packets to reach nodes at the maximum number of hops from the CH.

Sample Schedule: A sample schedule generated by the proposed scheduler is given in Figure 34. Each column is a slot; we show only 12 slots, which is a partial frame. In the first column the node's UIDs identify the CCs in Figure 9. Subsequent columns indicate the VID of the sending and the receiving nodes with arrows showing the direction of transmission. For example, in slot 1 CH VID '1' sends to node B VID '11'. The slot allocation process proceeds by allocating slots from the CH to one-hop nodes, followed by the one-hop CCs sending to their two-hop children and the two-hop CCs sending to their three-hop children and so on. However, due to the directional antennas used, we can have simultaneous transmission between two pairs of distinct nodes, for example, in slot 3 CH is sending to node D on VID '13', but node B using VID '11' is sending to node A at VID '111'. A closer look at the latter half of the schedule will reveal that the flow from the outer leaf nodes to the CH is the mirror of the allocation process from CH to leaf nodes, i.e., the 1st hop children are allocated the last time slots in the frame, thus allowing for data aggregation as the packets are moved towards the CH.

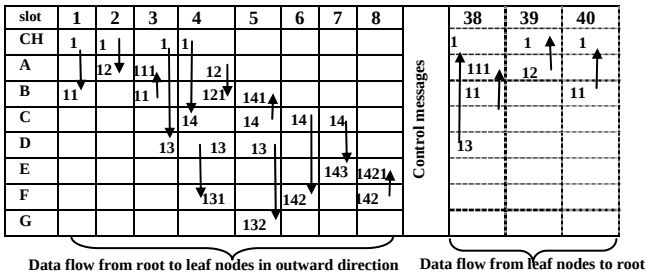

Figure 34 Sample Schedule in Meshed Tree Clusters

We conducted simulations using Opnet for 20-, 50- and 75-node multi-hop networks. All nodes were randomly assigned clockwise and counter-clockwise circular trajectories with 100 km radii and speeds varying from 200, 250, 300, and 350 km/h at an altitude of 20 km.

Frames with 28 slots each were used for the 20- and 50-node scenario. 42 slots were employed for the 75-node scenario because the meshing in the cluster results in most nodes using up all 6 VIDs, thus requiring additional slots. Each slot had a 12.5ms duration and a guard time of 1.5 ms. The cluster size was maintained at 12 with a maximum of distance of three hops between CC and CH. Nodes were allowed to have a maximum of six VIDs.

Nodes in the network were randomly selected to send 1 MB file simultaneously in 2 kB packets to randomly selected destination nodes. Overhead, average hops, successful packet delivery rate, as well as mean packet latencies were measured, where

- *Success rate* was calculated as the number of packets delivered to the destination node successfully as a percentage of the number of packets that originated at the sender node

• *Overhead* was calculated as the ratio of control bits to the sum of control and data bits during data delivery.

Each simulation was run with several seeds and the average values were plotted in the graphs shown in Figures 35 - 37. As there are no published results for such network scenarios to the best of our knowledge, we use the graphs to highlight the performance of the proposed solution.

20-Node Scenario: The number of sending nodes was varied from 4 to 8 to 16 nodes. In the case of 16 senders, all CCs sent 1 MB files to all other CCs in the network; this is the stress test case.

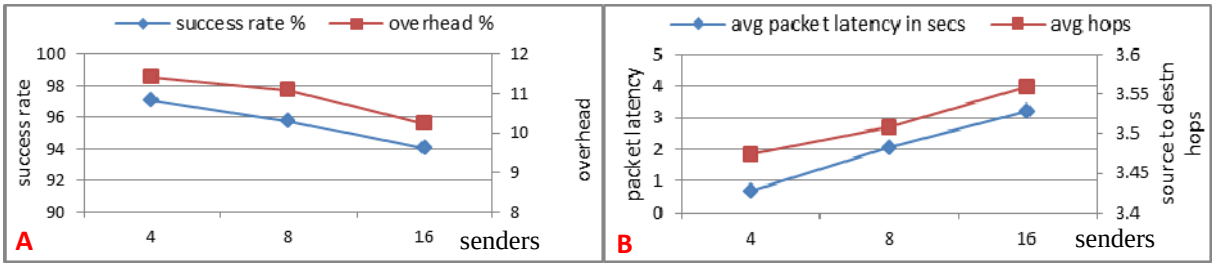

Figure 35 Performance Graphs – 20-Node Scenario (airborne backbone/directional)

As shown in Figure 35A, the success rate was around 97% with 4 senders and dropped to 94% with all 16 senders. As traffic in the network increases the overhead is reduced; this is due to the inverse relationship between the traffic load in the network with respect to the control bits generated. In Figure 35B, the average packet latency increased from 0.7 second to 3 seconds when the traffic in the network increased. We recorded the average hops encountered between sending and receiving nodes to track the affect the distance between the communicating nodes has on the network success rate and overhead. In this case the recorded average number of hops was around 3.5 hops.

50-Node Scenario: In this network scenario, the number of simultaneously sending nodes was varied from 10 to 20 to 40. There were 10 CHs in this scenario, thus in the case of 40 sending nodes all CCs were sending to all other CCs within the network. With 10 sending nodes, the success rate was 94%, dropping to around 83% when 40 were used. The 23% overhead is higher than that for the 20-node scenario due to the increase in route discovery maintenance across 50 nodes. The average packet latency was 2.5 seconds with 10 senders and 11 seconds with 30 senders due to the increase in network traffic associated with the larger number of sending nodes. The average hops recorded were between 6 and 7.

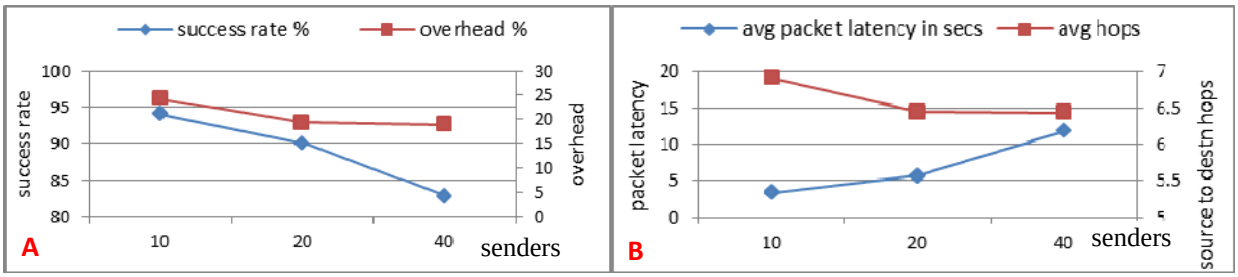

Figure 36 Performance Graphs – 50-Node Scenario (airborne backbone/directional)

75-Node Scenario: In this scenario we varied the number of senders from 15 to 30 to 60 nodes. Using 15 CHs, all nodes were again sending traffic to all other destination nodes. The 90% success rate for 15 senders dropped to 70% when all 60 senders were present. The overhead varied from 27 to 30% and an increase in average packet latency due to the increased network traffic is clearly visible.

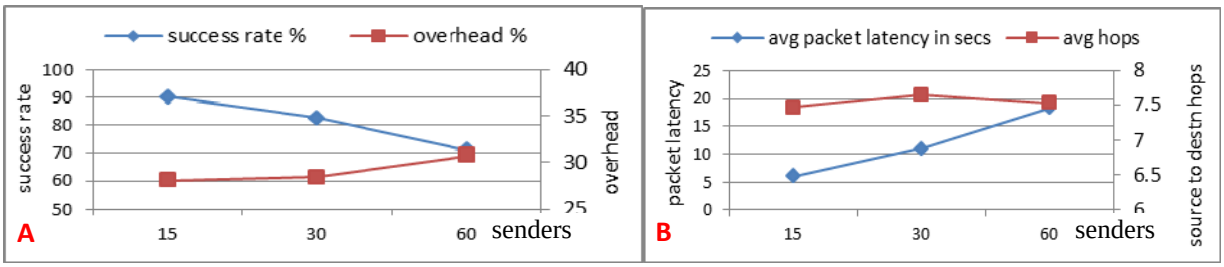

Figure 37 Performance Graphs – 75-Node Scenario (airborne backbone/directional)

Summary of Results: A high value of successfully delivered packets with some acceptable latencies based on the traffic in the network were observed. In general, high success rate is difficult to achieve in such highly dynamic MANETs, especially when the number of mobile nodes is also high (several tens in our case). This performance assessment is appropriate if the data type represents simple data files.

4.5.2 Distributed Scheduler and Inter-Cluster Communications

A distributed TDM scheduling algorithm allows nodes in the cluster to schedule their own transmission and reception slots with their neighbors and uses the cluster formation process to determine when nodes join to establish new link schedules.

Inter-Cluster Reactive Routing: The meshed tree-based cluster formation allows nodes to join different clusters. CCs which are in multiple clusters will have VIDs from each cluster the first digit of which can be used to identify the cluster to which they belong. Also, since all CCs register their multiple VIDs with their CHs, the CHs are aware of the neighboring clusters and their respective CH IDs. This information will enable the CCs to resolve any conflicts when scheduling time slots among different clusters. Meanwhile the CHs will use the border node information to propagate the route discovery messages. Also, by allowing nodes to belong to multiple clusters the single meshed tree cluster is extended to *multiple* overlapping *meshed tree* (MMT) clusters covering a wider area and thus addressing *scalability.*

Two overlapped clusters with CH 1 and CH 2 created this way are shown in Figure 38. Notice the multiple VIDs under different clusters for nodes F and G, where the first digit (as we used single digit IDs) is the ID of the CH. A node that has to discover a route to a distant node (a node that is not its one-hop neighbor) sends a 'route request' message to its CH(s); for example, in Figure 38, if node L would like a route to node B, it will forward the route request message to CH 2. CH 2 in this case knows that it has one neighbor cluster CH 1. From its list of registered VIDs it will identify nodes G and F to be border nodes that connect CH 1 and CH 2. CH 2 will forward a copy of the route request to the border node with the shortest hops. In this case, however, as both nodes are 2 hops from CH 2, CH 2 can choose either of the border nodes to send the route request message. Let us assume that the route request is forwarded to node G. Node G will then forward the route request message to CH 1. CH 1 will identify node B within its cluster, use one of its VIDs (the shortest one), and forward the route request to B.

To avoid message looping, route request messages sent by a CH will have an entry for all the clusters receiving the message. The CHs forwarding the route request message also record the original sending node and the last CH that the route request came from; this information is useful in forwarding the *route response* message when it returns.

The destination node generates the *route response* and sends to its CH which in this case is CH 2. CH 2 will then forward the route response message back to CH 1 which is the originating cluster and the source node along the same *cluster path* the *route request* message took. Along the path back, all forwarding CHs will record the previous CH and original sender of the route response message. The route between the sender and the destination node is thus initially set up as a sequence of CHs but maintained as next CH information.

Reactive Routes: Note in Figure 38 that we have used thick arrows to show the path taken by the route request message from L-CH2-CH1-B, via nodes H, G, and D, while the *route response* message from B-CH1-CH2-L via the nodes C, F and H. This shows that the reactive route, which is a sequence of CHs, has no dependency on the intermediate nodes that forward the request and response messages. The same holds true for data packets when they are forwarded using reactive routes; the probability of reactive route failure is thus considerably reduced. Reactive routes are concatenations of the current proactive routes existing in the cluster. Mobility of nodes does not impact

the reactively discovered route as long as the CHs exist. Note that movement of CHs also does not impact the reactive routes.

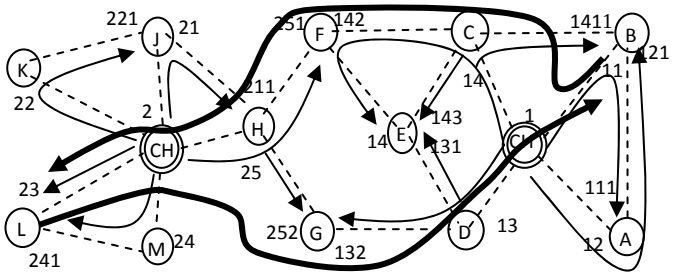

Figure 38 Reactive Routing-based on Meshed Trees Clusters

The Distributed Scheduler: We assume that all nodes are aware of the frame size, slot size, and guard time. Also, because of the GPS, we further assume that nodes are synchronized to the frame and slot start and end times. We categorize the slots into three types: control slots, data slots and temporary (temp) slots. The functions of each of these slot types will now be described.

Control Slots: For our scenarios we used two control slots per node per frame. These slots are randomly assigned when each node is initialized. One of the slots is used for broadcasting messages and the other only for receiving. During the transmit slot, the sending node chooses a random start time that will allow enough time left in the slot to fully send its packet. Nodes in range do not have to have a matching receive control slot, they can receive the packet on any temp slot or on any slot in which they are not actively transmitting. These slots are used by nodes to advertise their VIDs and broadcast information using *configuration* packets. When advertising using such control slots, nodes that are in listening mode pick up the advertisements as they are sent by a node in all quadrants using a defocused beam.

Data Slots: These slots are also categorized as data-*tx* slots and data-*rx* slots. These slots are used for transmitting and receiving all data and most routing packets between neighboring nodes using focused beams. The MAC in the nodes also uses these slots to send *Link Maintenance* packets which are prioritized before data. These packets keep neighboring nodes that have a link (via an acquired VID) active. In the event of three missing link maintenance packets, the MAC decides the link is broken and informs the MMT clustering and routing layers. The VID will then be removed from the list.

Temp Slots: These are slots that have not been set aside either for control purposes or scheduled for sending and receiving data. A pair of nodes may use these slots to communicate with one another specifically when either desires to join the cluster and acquire a VID under a given parent node. When a node does not have packets to transmit on a temp slot it is in the listening mode. Temp slots may eventually become data-*tx* or data-*rx* slots.

Data Slot Allocation: The data-*tx* and data-*rx* slot allocation is done in the following manner: When a node advertises its VIDs via a configuration packet, it also attaches its schedule and GPS coordinates. The node that receives the advertisement, if it decides to request a VID under the advertised VID, will then allocate a data-*rx* slot from one of the temp slots advertised by the parent that matches its own temp slot. It will then send a registration request with the new schedule to the parent. The parent in turn assigns another temp slot as a data-*rx* slot for receiving packets from the child and forwards the registration request towards the CH. During the next frame, the parent will send a *Link Maintenance* packet to its child containing the new schedule.

Thus two nodes A and B have a set of mutually reserved slots for transmitting and receiving. Note that no other node's schedule is taken into account unless it directly affects the current link between two negotiating nodes. For example, node A cannot send to node B and receive from node C at the same time. Slot reuse is automatically maximized with this scheme and links more than two hops away can never affect the schedule decided by a pair of nodes. This is only possible due to the highly directional antennas which beam the packets to the destination nodes.

Conflict Resolution: There exists the possibility that two nodes may attempt to schedule the same slot for a third node. The third node will accept the data-*rx* allocation from the first schedule that it receives; when it gets the

second schedule, it will simply ignore it and send a link maintenance packet to the sender. The denied node will see the conflict and choose a different slot to allocate as data-*rx* slot. This is why nodes do not choose their data-*tx* slots; a node will not send traffic until the receiving node has set aside its own data-*rx*, so both sides of the link will be fully aware of the transmission.

Medium Access Control: The MAC handles all packet forwarding and antenna beam control. The MAC has multiple queues for this purpose. One queue is used for each next hop destination and stores all packets that have to be transmitted to that destination. Each temp slot has its own queue as well. The reason for this is that the receiver must have a temp slot at the same time or it might not be listening, so packets that will be sent on a temp slot must be sent on the specific slot that was chosen.

Table 3 Sample Schedule Generated by the Distributed

Slot	1	2	3	4	5	6	7	8	9	10
CH	TX to B	RX to B	CTRL	TX to A	RX to A	TX to D	RX to D	TEMP	TX to C	RX to C
A	TEMP	CTRL	TX to B	RX to CH	TX to CH	RX to B	TEMP	TEMP	TEMP	TEMP
B	RX to CH	TX to CH	RX to A	TX to C	RX to C	TX to A	TEMP	CTRL	TEMP	TEMP
C	TX to E	TX to F	RX to E	RX to B	TX to B	CTRL	RX to F	TEMP	RX to CH	TX to CH
D	CTRL	TX to E	TX to G	RX to E	RX to G	RX to CH	TX to CH	TEMP	TEMP	TEMP
E	RX to C	RX to D	TX to C	TX to D	CTRL	TEMP	TEMP	RX to F	TX to F	TEMP
F	TEMP	RX to C	CTRL	TEMP	TEMP	TEMP	TX to C	TX to E	RX to E	CTEL
G	TEMP	TEMP	RX to D	TEMP	TX to G	TEMP	TEMP	TEMP	CTRL	TEMP

Acknowledgements (ACK): In the proposed scheme, we acknowledge route requests, route replies, and data packets only. Each ACK contains a low and a high sequence number which represent the range of packets that are being acknowledged. If the sender of the packet does not receive the corresponding ACK by the following frame, it will attempt to resend the packet up to a maximum of three attempts. At that point, the MAC will decide that the link has failed and the VID will no longer be valid. It will inform the MMT process to remove the VID. Any queued packets will be rerouted.

Sample Schedule: Table 3 is a sample schedule generated for the cluster in Figure 2. Nodes A, B, C, and D receive the initial configuration packet from the cluster head and schedule their data-*rx* (RX) slots; 4, 1, 9, and 6 respectively. This decision is a random allocation of matching temp slots based on the sequence in which the configuration packets are received. The cluster head accepts these data-*rx* packets which were sent in the registration request messages of these nodes and sets the corresponding slots as data-*tx* (TX) slots in its own schedule. It then allocates data-*rx* slots to each of these nodes on slots 5, 2, 10, and 7. Node A receives a configuration packet from Node B and decides to use slot 6 as its data-*rx* slot for receiving from node B. Node B then chooses slot 3 as the data-*rx* slot for A. At the same time Node B receives Node C's configuration and chooses slot 5 as its data-*rx* slot. Node C selects slot 4 as the complementary slot. This is the same slot that the CH is transmitting to Node A, but due to the directional antennas there will be no interference. The process continues branching outward until every link has a pair of slots allocated.

Simulation Results: We conducted simulations using Opnet for 20-, 50- and 75-node multi-hop networks. All nodes were randomly assigned clockwise and counter-clockwise circular trajectories, with a 100 km radius and speeds varying between 250, 300, and 350 m/h at an altitude of 20 km. The circular trajectories provide a stressful test as they result in many route breaks.

The frames had 28 slots each for every scenario. Each slot had a 12.5 ms duration and a guard time of 1 ms. The maximum cluster size was 12 nodes with a maximum of a three-hop distance between a CC and CH, and the most VIDs a node could have was set at 6. Nodes in the network were randomly selected to send 1 MB file simultaneously in 2 kB packets to randomly selected destination nodes. Overhead, average hops, successful packet delivery rate, as well as mean packet latencies were measured where:

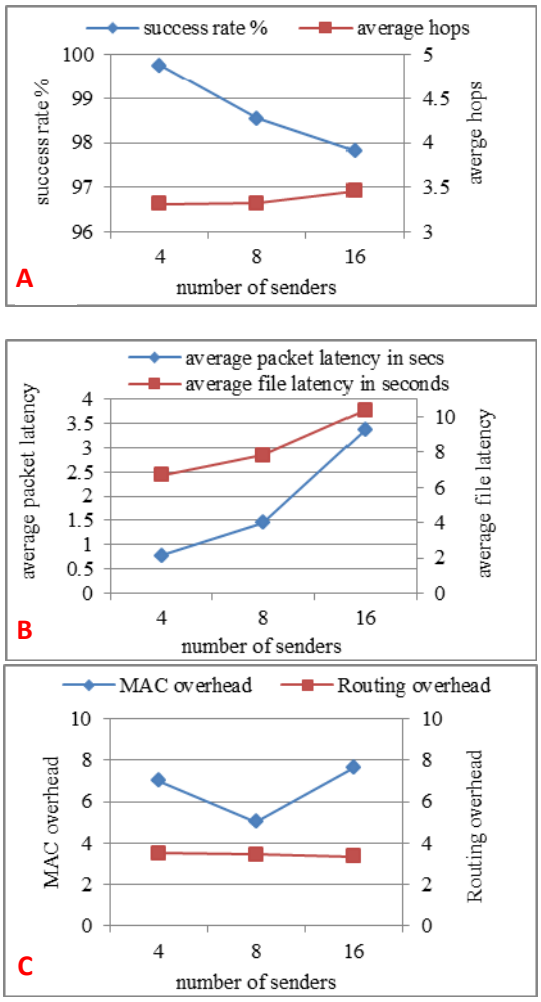

Figure 39 Performance graphs – 20-Node Scenario (airborne Backbone/distributed scheduler)

- *Success rate* was calculated as the number of packets delivered to the destination node successfully as a percentage of the number of packets that originated at the sender node
- *Overhead* was calculated as the ratio of control bits to the sum of control and data bits during data delivery.

Each simulation was run with several seeds and the average values were plotted in the graphs shown in Figures 39-41. As there are no published results for such network scenarios to the best of our knowledge, we use the graphs to highlight the performance of the proposed solution.

20-Node Scenario: Figures 39A-C are the performance plots for the 20 airborne node scenario. On the *x*-axis we plot the number of nodes that are sending traffic which were selected randomly. The destination nodes were also selected randomly.

The number of sending nodes was varied from 4 to 8 to 16 nodes. In the case of 16 senders, all CCs were sending 1 MByte files to all other CCs in the network, which is stress test case, and thus will be rarely encountered in a real world network. The setup that allows all nodes to send 1 Mbyte file simultaneously also imposes stress on the network algorithms. We specifically considered such stressful scenarios to highlight the robustness of the proposed scheme which is essential if it will be used for critical applications such as airborne backbone networks for interconnecting tactical sub-networks.

As the number of senders was increased, the success rate dropped from 99.75% to 97.7%, which is very low. In the case of the 16 traffic senders, all nodes are communicating to all other nodes and traffic in the network is very high. The average hops were plotted to show a typical number of hops that the packets traversed as they were forwarded between the source and destination nodes. The slight increase in the average hops at 16 senders indicates that as the sending and destination nodes were randomly selected, in the case of the 16 sender scenario, these nodes were further apart than the pairs selected for the 4 and 8 sender scenario.

The average packet delivery latency is around 0.7 seconds with 4 senders and goes up to 3.5 seconds with 16 senders. This latency increase with a higher number of senders is because of the high traffic in the network. The average file latencies also increased from 6.5 seconds with 4 senders to 11 seconds with 16 senders.

Figure 39C shows an interesting graph on the routing and MAC overhead. The routing overhead is quite low as compared to the MAC overhead at almost half the value and was stable as the number of senders was increased. This is so because of the way that the overhead is calculated, i.e., overhead is the ratio of the sum of control bits to the total number of bits in the network when the data is being sent. Hence, with a higher number of sending nodes, the overhead should show a decline. However, one notices that the MAC overhead has increased with the increase in the number of senders which is due to the higher number of average hops encountered with 16 senders. The MAC overhead is also higher than the routing overhead as the MAC issues link maintenance packets when there is no data to send. It also has to resolve conflicts.

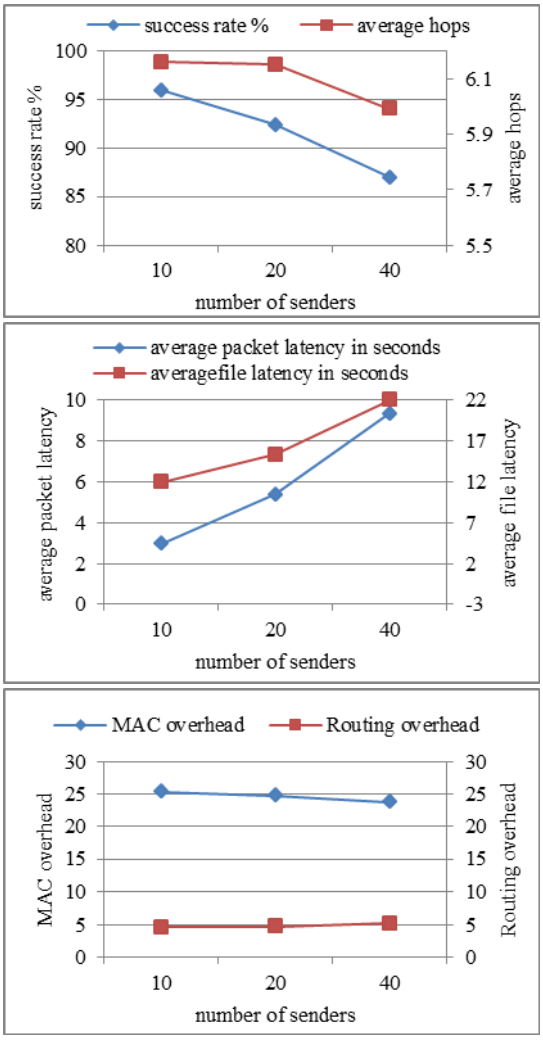

Figure 40 Performance Graphs – 50-Node Scenario (airborne backbone/distributed scheduler)

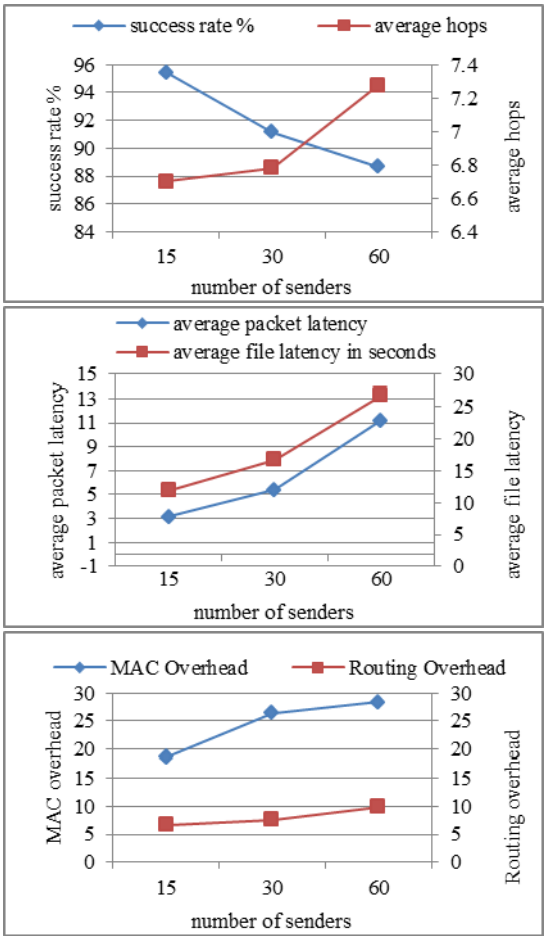

Figure 41 Performance Graphs – 75-Node Scenario (airborne backbone/distributed scheduler)

50-Node Scenario: In this network scenario, the number of simultaneously sending nodes was varied from 10 to 20 to 40. There were 10 CHs in this scenario, hence with 40 sending nodes all CCs were sending to all other CCs in the network. The arguments stated for the 20-node scenario also apply to this scenario.

From Figure 40A, the success rate can be observed to decline more steeply from 96% with 10 senders to 87% with 40 senders, which is logical to expect with the increased traffic, especially when all nodes are sending to all other nodes in the network.

The average packet latency with 10 senders is around 3 seconds and goes to 9 seconds with 40 senders. This is attributed both to the higher number of senders in this scenario as compared to the equivalent scenario of 4 senders in 20 nodes and also to the higher number of hops that the nodes encountered, which is around 6. However, note that due to the random selection of sending and receiving node pairs, the average hops with 40 senders is now slightly lower than with 10 and 20 senders. The impact on the overhead is noticed in Figure 40C where the MAC overhead actually shows a decline. The routing overhead continues to be stable around 5%.

75-Node Scenario: In this scenario we varied the number of senders from 15 to 30 to 60 nodes. As the number of CHs was 15, in this case again we had all nodes sending traffic to all other destination nodes.

The performance graphs for the 75-node scenario are shown in Figures 41A-C. A typical trend similar to that noticed with the 20-node scenario and 50-node scenario is observed. The success rates dropped from 95% with 15

senders to around 88% with 60 senders. The average hops are between 6 and 7, slightly higher than that in the 50-node scenario; this is because the 25 new nodes were placed around the periphery of the 50-node scenario.

There is an increase in the MAC overhead with increasing senders due in part to the fact that the average hops increased and also the reduced network traffic. The routing overhead exhibits a slight increase from 6% to 10% with increased senders for the same reasons.

Summary of Results: We have evaluated our algorithms and solution under highly stressful scenarios to emphasize the effectiveness and robustness of the algorithm. The low routing overhead and its maintenance of a stable value to changing numbers of senders indicate the strength of the reactive routing approach. Included in this routing overhead are the control bits for maintaining the proactive routes within the cluster and the reactive route discovery. The highly predictable trends are also indicative of the stability of the algorithms and the models.

The solution is unique both from the perspective of the TDMA scheduler and the reactive routing protocol. The preliminary evaluations of this scheme show the very promising results that were obtained for airborne backbone networks. The consistent performance is also indicative of the stability and robustness of the proposed algorithms.

- Physical layer was modeled using Opnet available models but not with the facilities available at AFRL. Doppler effect was not modeled.
- The MMT algorithm has been modeled to take parameters form the physical layer and applications layer but these have not yet been tested under a cognitive paradigm.

5 Conclusions

The proposed work involved design for a compact protocol stack with routing and MAC functions and gateways capable of addressing integration across heterogeneous networks and an evaluation of the proposed design and its capability to handle quality-sensitive application. The protocol stack was developed and integrated solutions were investigated using Opnet for various airborne and ground network scenarios. The performance results indicate the capability of the solution to address heterogeneous networks. The study for different quality-sensitive applications was limited to support of text files of small size in the case of ground troops, 5 to 10 kB, and large, 1MB, data files in airborne networks.

6 Bibliography

1. Haas, Z.J.; Tabrizi, S., "On some challenges and design choices in ad-hoc communications," *Military Communications Conference, 1998. MILCOM 98. Proceedings., IEEE* , vol.1, pp.187-192 Oct 1998
2. Sanchez, R.; Evans, J.; Minden, G., "Networking on the battlefield: challenges in highly dynamic multi-hop wireless networks," *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE* , vol.2, no., pp.751-755 vol.2, 1999
3. Ramanathan, R.; Redi, J., "A brief overview of ad hoc networks: challenges and directions," *Communications Magazine, IEEE* , vol.40, no.5, pp.20-22, May 2002
4. Gerla M., "From battlefields to urban grids: new research challenges in ad hoc wireless networks," *Pervasive Mobile Computing*, vol. 1, pp. 77-93, 2005
5. Bandyopadhyay S.; Coyle, E.J., "An energy efficient hierarchical clustering algorithm for wireless sensor networks," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE* , vol.3, no., pp. 1713-1723 vol.3, 30 March-3 April 2003
6. Shenoy N., Yin Pan, Vishal Gogula Reddy, "Quality of Service in Internet MANETs," Invited paper, *16th International Symposium on Personal Indoor and Mobile Radio Communications PIMRC 2005*. Sept 11-14 2005. International Congress Center, Berlin, Germany
7. Shenoy N., Yin Pan, "Multi-Meshed Tree routing for Internet MANETs," *Proceedings of the 2nd IEEE International Symposium on Wireless Communications Systems*, Sept 5-7 2005, Sienna, Italy.
8. Shenoy N., Yin Pan, Darren Narayan, David Ross, Carl Lutzer, "Route Robustness of a Multi-Meshed Tree Routing Scheme for Internet MANETs," *Proceedings of the IEEE Globecom 2005*. 28 Nov – 2nd Dec. 2005 St Louis.
9. Shenoy N., Pan Y., Reddy V. G., "Bandwidth Reservation and QoS in Internet MANETs," *Fourth IEEE International Conference on Computer Communications and Networks*, Oct 17-19 2005. San Diego, USA.
10. Pudlewski S., Shenoy N., Al Mousa Y., "A Hybrid Multi-Meshed Tree Routing protocol wireless ad hoc networks," *Second IEEE International Workshop on* Enabling Technologies and Standards for Wireless Mesh Networking, September 29, 2008. Atlanta, GA, USA
11. Qin L. Kunz T., *Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design*, Technical Report Systems and Computer Engineering, Carleton University, August 2004
12. Abolhasan M., Wysocki T., Dutkiewicz E., "A review of routing protocols for mobile ad hoc networks," *Journal of Ad Hoc Networks*, Elsevier publications, 2004
13. Daniel L., *A comprehensive overview about selected Ad hoc networking routing protocols*, Technical Report, Department of Computer Science, Technische Universitat, Munchen, Germany
14. Royer E. M., C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications Magazine*, April 1999, pages 46-55.
15. Perkins C., E., E. M. Royer, and S. R. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, IETF Mobile Ad Hoc Networks Working Group, IETF RFC 3561
16. Johnson D. B., D. A. Maltz, Y-C Hu., *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, IETF Mobile Ad Hoc Networks Working Group, Internet Draft, 24 February 2003
17. Perkins C. E., P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Proceedings of the ACM Special Interest Group on Data Communications (SIGCOMM)*, August 1994, pages 234-244
18. Clausen T., Ed., P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, Network Working Group, Request for Comments: 3626
19. Broch J., D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM)*, October 1998, pages 85-97
20. Das S., R. Castaneda, J. Yan, "Simulation-Based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications*, 2000, Vol. 5, No. 3, pages 179-189
21. Baker D.; Ephremides A., "Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *Communications, IEEE Transactions on*, vol.29, no.11, pp. 1694-1701, Nov 1981
22. Basagni S.; Chlamtac I.; Farago A., "A generalized clustering algorithm for peer-to-peer networks," *Workshop on Algorithmic Aspects of Communication*, July 1997
23. Park V.D., M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *Proceedings of INFOCOM*, April 1997.

24. Hong X., Kaixin Xu, Mario Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network Journal*, July/Aug 2002, Vol 16, issue 4, pages 11-2.
25. Iwata A., C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad-hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Aug. 1999, pp. 1369-1379.
26. Pei G., M. Gerla, and T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *IEEE International Conference on Communications*, 2000, Vol 1, pages 70-74.
27. Bellur B. and R. G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," in *Proceedings of IEEE INFOCOM '99*, New York, March 1999.
28. Qayyum A., L. Viennot, A. Laouiti, *Multipoint relaying: An efficient technique for flooding in mobile wireless networks*, INRIA research report RR-3898, 2000
29. Santivanez C., R. Ramanathan, I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks," *Proceedings of The 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc2001)*, Long Beach, California, Oct. 2001.
30. Das S.R., C.E. Perkins, E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.
31. Chiang C. -C. and M. Gerla, "Routing and Multicast in Multihop, Mobile Wireless Networks," *Proceedings of IEEE ICUPC'97*, San Diego, CA, Oct. 1997.
32. Pei G., M. Gerla, X. Hong, and C. -C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sept. 1999.
33. Haas Z.J. and M.R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," *ACM/IEEE Transactions on Networking*, vol. 9, no. 4, August 2001, pp. 427-438.
34. Pei G., M. Gerla, X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," *Proceedings of IEEE/ACM MobiHOC 2000*, Boston, MA, Aug. 2000, pp. 11-18.
35. Gerla M., X. Hong, G. Pei, "Landmark Routing for Large Ad Hoc Wireless Networks," *Proceedings of IEEE GLOBECOM 2000*, San Francisco, CA, Nov. 2000.
36. Xu K., Hong, X., Gerla M., "An Ad hoc Network with Mobile Backbone," *ICC 2002 IEEE International Conference on Communications*, Volume 5, April, 2002 Page :3138 - 3143 vol.5
37. Ramasubramanian V., Haas Z. J., Emin G¨un Sirer, "SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks," *MobiHoc'03*, June 1–3, 2003, Annapolis, Maryland, USA.
38. Nasipuri A., Burleson R., Hughes B., Roberts J., "Performance of a Hybrid Routing Protocol for Mobile Ad Hoc Networks," http://www.coe.uncc.edu/~anasipur/pubs/hybrid.pdf
39. Belding-Royer E. M., "Multi-level Hierarchies for Scalable Ad hoc Routing," *Wireless Networking (WINET)*, Vol. 9, No. 5, pages 461-478, Sept. 2003
40. Bandyopadhyay S.; Coyle, E.J., "An energy efficient hierarchical clustering algorithm for wireless sensor networks," *INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE* , vol.3, no., pp. 1713-1723 vol.3, 30 March-3 April 2003
41. Younis, O.;Fahmy, S., "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* , vol.1, no., pp.-640, 7-11 March 2004
42. Chen Y.; Liestman A., Liu J., "Clustering Algorithms for Ad Hoc Wireless Networks," *Ad Hoc and Sensor Networks*, 2004
43. Chen G.; Stojmenovic I., "Clustering and routing in mobile wireless networks," *Technical Report TR-99-05, SITE*, June 1999
44. Gerla M.; Tsai J., "Multicluster, mobile, multimedia radio network," *Wireless Networks.*, vol. 1, pp. 255-265, 1995
45. Lian, J.; Agnew, G.B.; Naik, S., "A variable degree-based clustering algorithm for networks," *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on* , vol., no., pp. 465-470, 20-22 Oct. 2003
46. Amis, A. D.; Prakash, R.; Vuong, T.H.P.; Huynh, D.T., "Max-min d-cluster formation in wireless ad hoc networks," *INFOCOM 2000 Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings. IEEE*, vol.1, no., pp.32-41 vol.1, 2000
47. Lin, C.R.; Gerla, M., "Adaptive clustering for mobile wireless networks," *Selected Areas in Communications, IEEE Journal on* , vol.15, no.7, pp.1265-1275, Sep 1997
48. Basagni, S., "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th* , vol.2, no., pp.889-893 vol.2, 1999

49. Opnet Simulation tool, www.opnet.com
50. Network Simulator 2, http://www.isi.edu/nsnam/ns/
51. R. Nelson, L. Kleinrock, "Spatial-TDMA: A collision-free multihop channel access protocol," *IEEE Transactions on Communications* 33 (1985) 934–944.
52. I. Martinez and J. Altuna, "Influence of directional antennas in STDMA ad hoc network schedule creation," *International Workshop on Wireless Ad-hoc Networks*, London, UK, 2005.
53. S. G. Fernandez, *On the performance of STDMA Link Scheduling and Switched Beamforming Antennas in Wireless Mesh Networks*, Master's thesis, King's College London, United Kingdom, 2009.
54. J. Grönkvist and A. Hansson, "Comparison between graph-based and interference-based STDMA scheduling," *MobiHoc*, 2001.
55. J. Grönkvist, "Traffic controlled spatial reuse TDMA inmulti-hop radio networks," *The 9th IEEE InternationalSymposium on Personal, Indoor and Mobile Radio Communications*, 1998, pp. 1203–1207.
56. J. Grönkvist, "Assignment methods for spatial reuse TDMA," *First Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2000, pp. 119–124.
57. J. Grönkvist, A. Hansson, J. Nilsson, "A comparison of access methods for multi-hop ad hoc radio networks," *IEEE Vehicular Technology Conference*, 2000, pp. 1435–1439.
58. A. Dhamdhere, J. Grönkvist, "Joint Node and Link Assignment in an STDMA Network," *Proceedings of the IEEE Vehicular Technology Conference*, 22-25 April 2007, pp. 1066 – 1070.
59. J. Grönkvist, "Novel Assignment Strategies for Spatial Reuse TDMA in Wireless Ad hoc Networks," *Wireless Networks*, Springer Netherlands, ISSN 1022-0038, vol. 12, no. 2, pp. 255 – 265, 2006.
60. J. Grönkvist, Jan Nilsson, and D. Yuan, "Throughput of optimal spatial reuse TDMA for wireless ad-hoc networks," *Proceedings of the VTC 2004-Spring*, Milan, Italy, May 2004.
61. Nirmala Shenoy, Yin Pan, Darren Narayan, David Ross, Carl Lutzer, "Route Robustness of a Multi-Meshed Tree Routing Scheme for Internet MANETs," *Proceeding of IEEE Globecom 2005*. 28 Nov – 2nd Dec. 2005 St Louis.

List of Acronyms

AN	Airborne Networks
AODV	Ad hoc On-demand Distance Vector
CC	Cluster Clients
CH	Cluster head
DSDV	Destination Sequence Distance Vector
DSR	Dynamic Source Routing
GPS	Global Positioning System
LMR	Lightweight Mobile Routing
MAC	Medium Access Control
MBN	Mobile Backbone Networks
MMT	Multi-Meshed Trees
NCTS	Not Clear to Send
OLSR	Optimized Link State Routing
QoS	Quality of Service
RREP	Route Response
RREQ	Route Request
STDMA	Spatial Time-Division Multiple Access
TDL	Tactical Data Links
TDMA	Time-Division Multiple Access
TORA	Temporally Ordered Routing Algorithm
ZRP	Zone Routing Protocol